US012783481B2

(12) **United States Patent**
Miles et al.

(10) **Patent No.: US 12,783,481 B2**
(45) **Date of Patent: Nov. 3, 2026**

(54) **DATABASE RETENTION AND DESTRUCTION SYSTEMS FOR RECORDS MANAGEMENT**

(71) Applicant: **Truist Bank**, Charlotte, NC (US)

(72) Inventors: **Anthony Patrick Miles**, Richmond, VA (US); **David Alan Vogeler**, Punta Gorda, FL (US); **Adesh Gupta**, Matthews, NC (US)

(73) Assignee: **TRUIST BANK**, Charlotte, NC (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **19/050,321**

(22) Filed: **Feb. 11, 2025**

(65) **Prior Publication Data**

US 2026/0236446 A1 Aug. 13, 2026

(51) **Int. Cl.**
G06F 16/00 (2019.01)
G06F 16/21 (2019.01)

(52) **U.S. Cl.**
CPC ***G06F 16/217*** (2019.01)

(58) **Field of Classification Search**
CPC G06F 16/217
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

7,216,115	B1	5/2007	Walters
10,187,542	B1	1/2019	Fielding
10,193,975	B2	1/2019	Beerana
10,303,805	B2	5/2019	Zhou
11,586,621	B1	2/2023	Aya
11,757,626	B1	9/2023	Rivlin
11,768,803	B2	9/2023	Jain
12,072,868	B1	8/2024	Opincariu
2003/0200223	A1	10/2003	Hack
2005/0055519	A1	3/2005	Stuart
2007/0177219	A1	8/2007	Shinozaki
2008/0104149	A1	5/2008	Vishniac
2013/0024429	A1	1/2013	Raas
2013/0304761	A1	11/2013	Redlich
2013/0339297	A1	12/2013	Chen
2013/0339303	A1	12/2013	Potter
2013/0339643	A1	12/2013	Tekade

(Continued)

FOREIGN PATENT DOCUMENTS

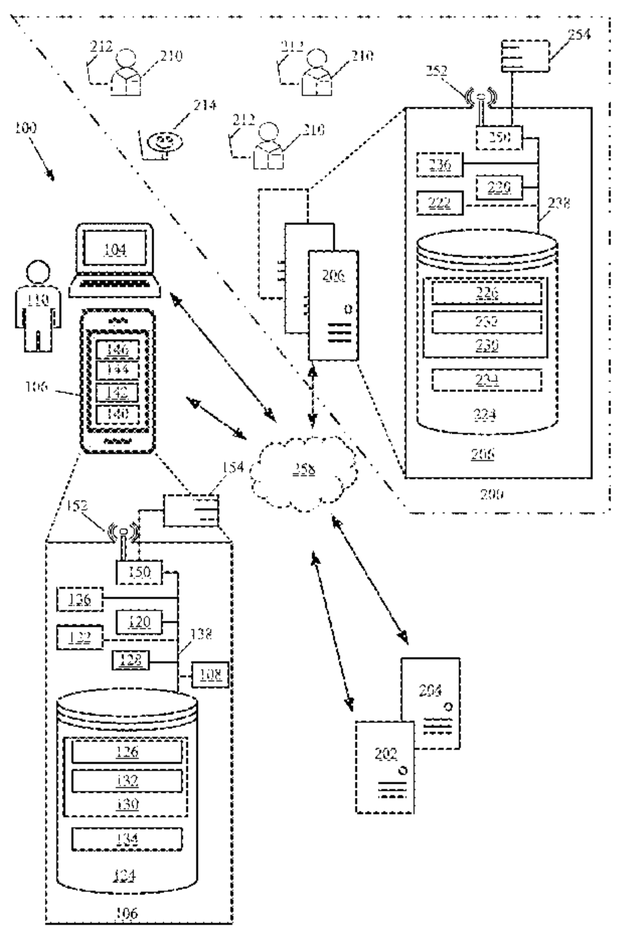
WO	2019190790	A1	10/2019

Primary Examiner — Paul Kim
(74) *Attorney, Agent, or Firm* — Michael A. Springs, Esq.; Shumaker, Loop & Kendrick, LLP; Patrick B. Horne

(57) **ABSTRACT**

Systems and methods initiate display of a module inventory interface including input(s) for data retention and destruction enrollment, at least one of the input(s) configured to define a retention period for object(s). An indication of a defined retention period for an object is received. Display of a required input for defining a field to be used for the defined retention period for the object is initiated, and a selection for the field to be used for the defined retention period for the object is received. Display of a retention lead input for indicating a records reviewer for authorizing destruction of the object outside of the defined retention period or retaining the object beyond the defined retention period and a name for the records reviewer is received. Display of a confirmation input for receiving confirmation of selections received via the module inventory interface is initiated, and the confirmation is received.

20 Claims, 36 Drawing Sheets



(56) **References Cited**

U.S. PATENT DOCUMENTS

2015/0310025	A1	10/2015	Rathgeber	
2017/0371890	A1	12/2017	Seker	
2018/0253218	A1	9/2018	Levesque	
2018/0365110	A1	12/2018	Venkatesh	
2019/0057101	A1	2/2019	Esserlieu	
2019/0102841	A1	4/2019	Nesanelis	
2021/0209251	A1	7/2021	Parthasarathy	
2022/0083505	A1	3/2022	Kushwah	
2022/0138048	A1	5/2022	Yelheri	
2023/0359597	A1*	11/2023	Bodkhe	G06F 21/6245
2024/0012797	A1*	1/2024	Jagasia	G06F 16/219
2025/0117371	A1*	4/2025	Li	G06F 16/125
2026/0023826	A1*	1/2026	Davis	G06F 21/105

* cited by examiner

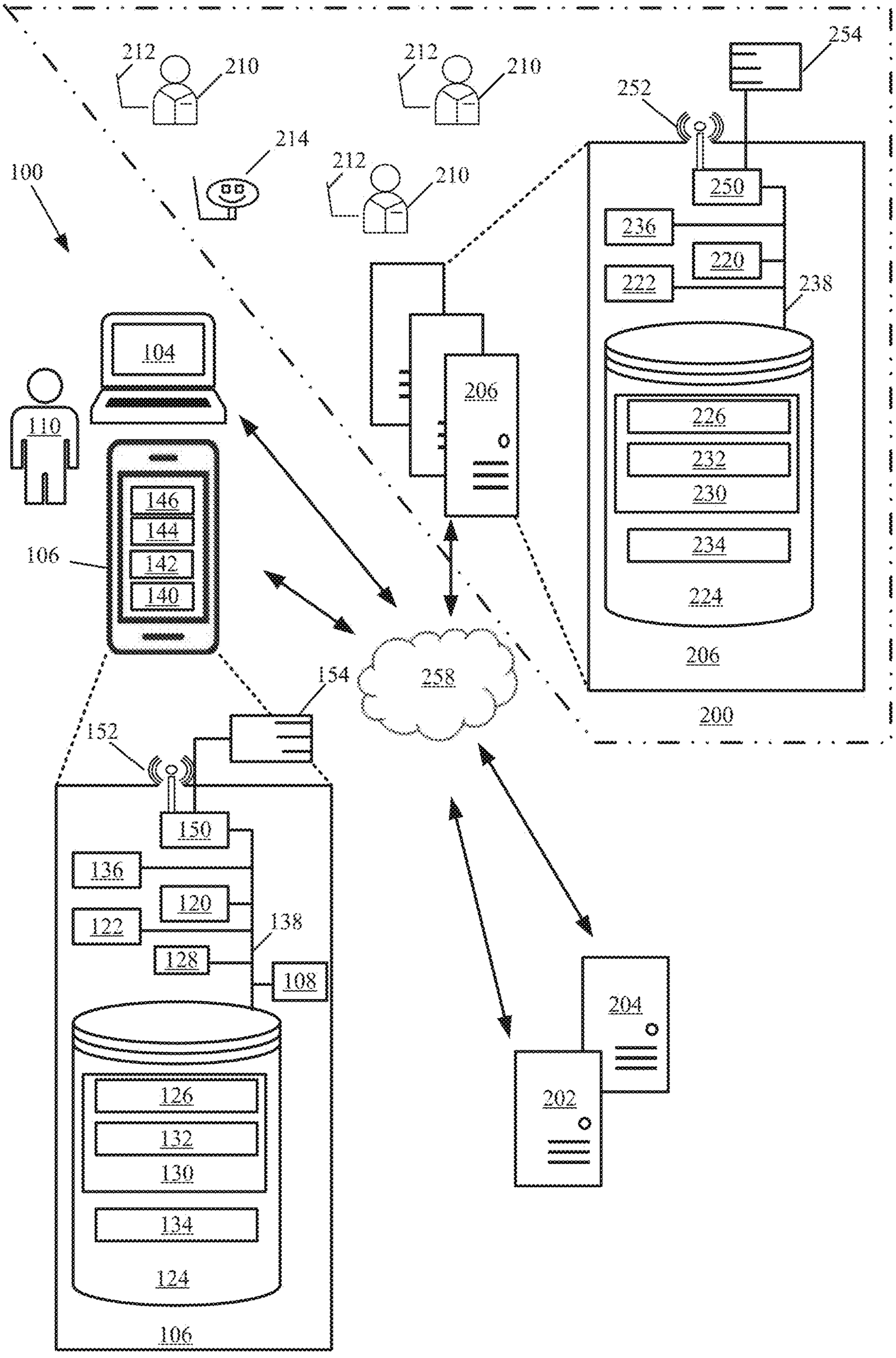

212
210
212
210
214
212
210
100
254
252
250
236
220
222
238
104
206
110
226
232
230
234
224
206
146
144
142
140
106
154
258
200
152
150
136
120
122
138
128
108
204
126
132
130
134
124
202
106

FIG. 1

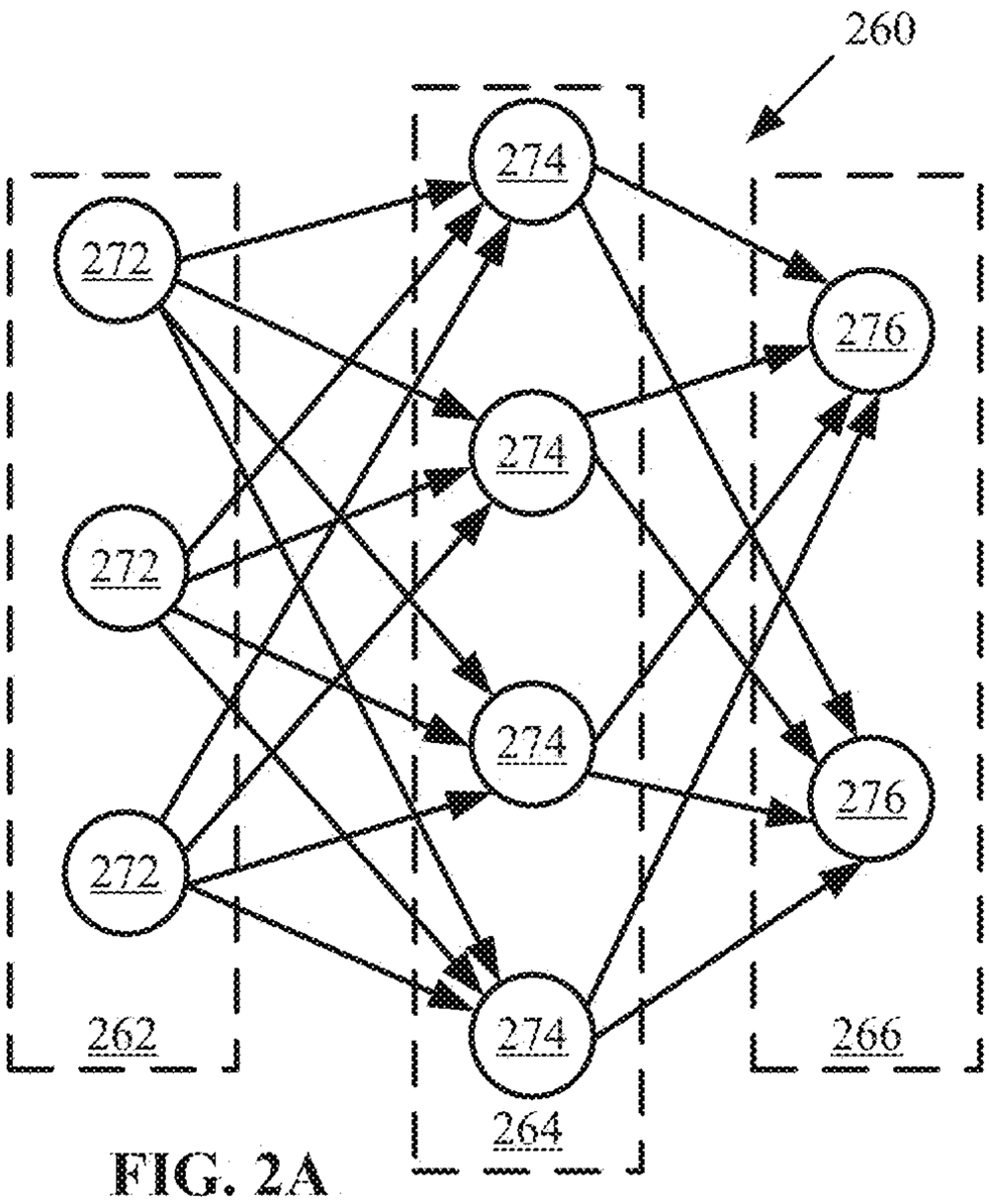


FIG. 2A

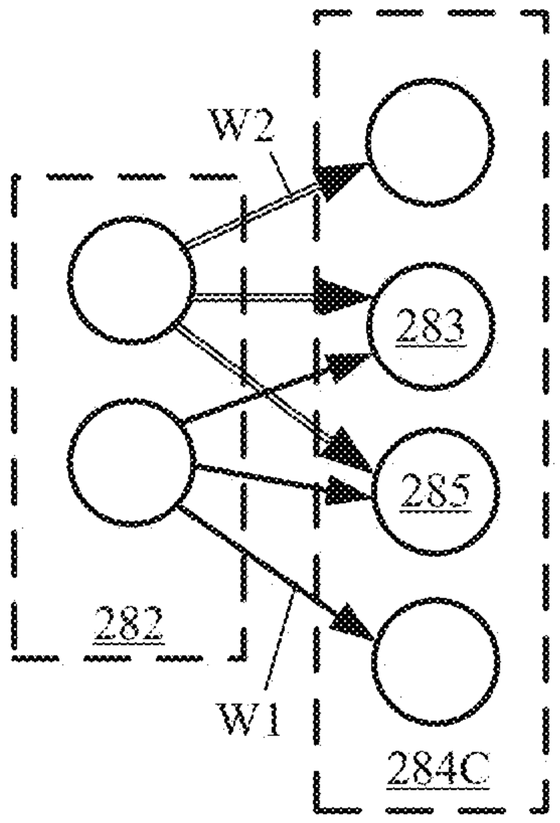


FIG. 2C

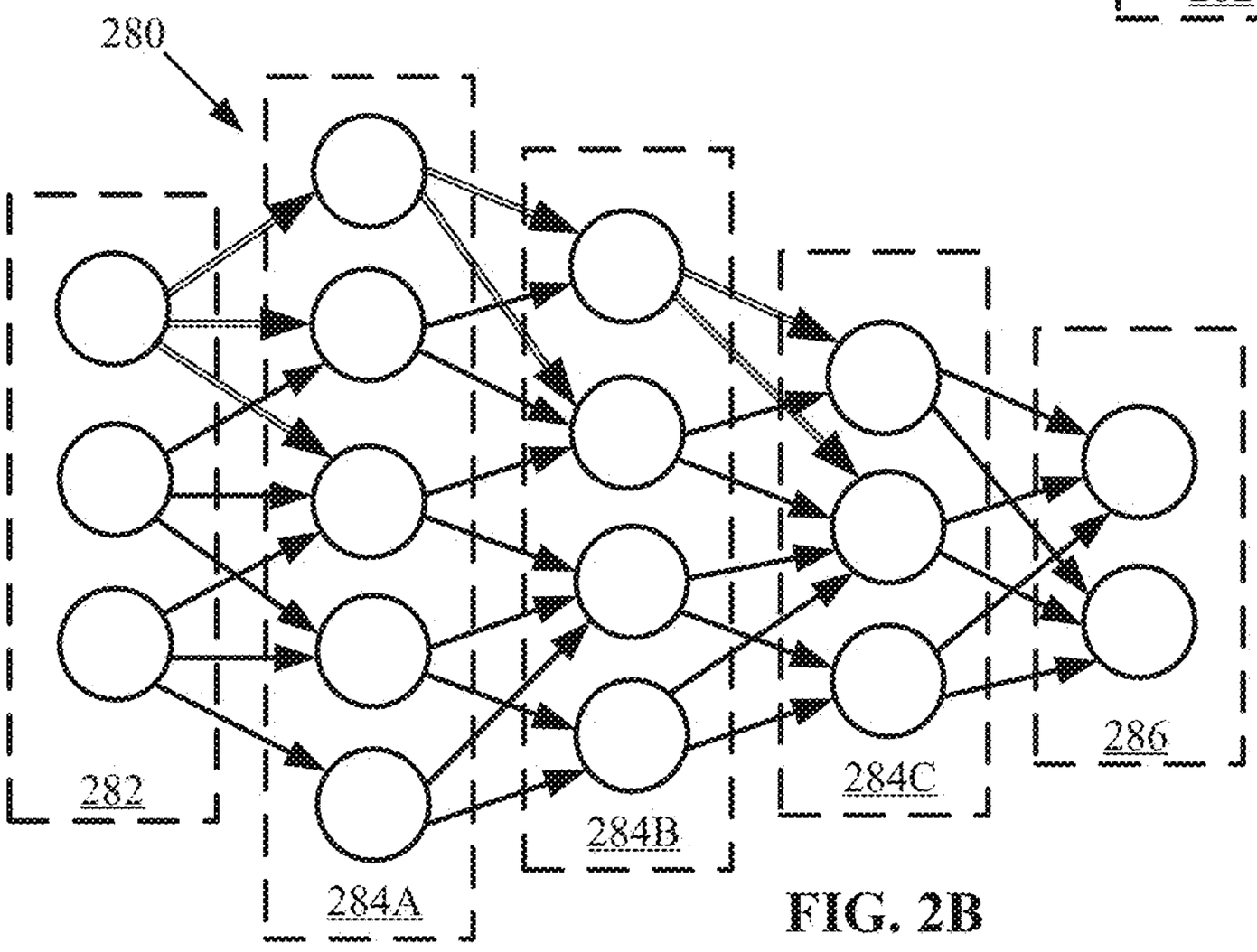


FIG. 2B

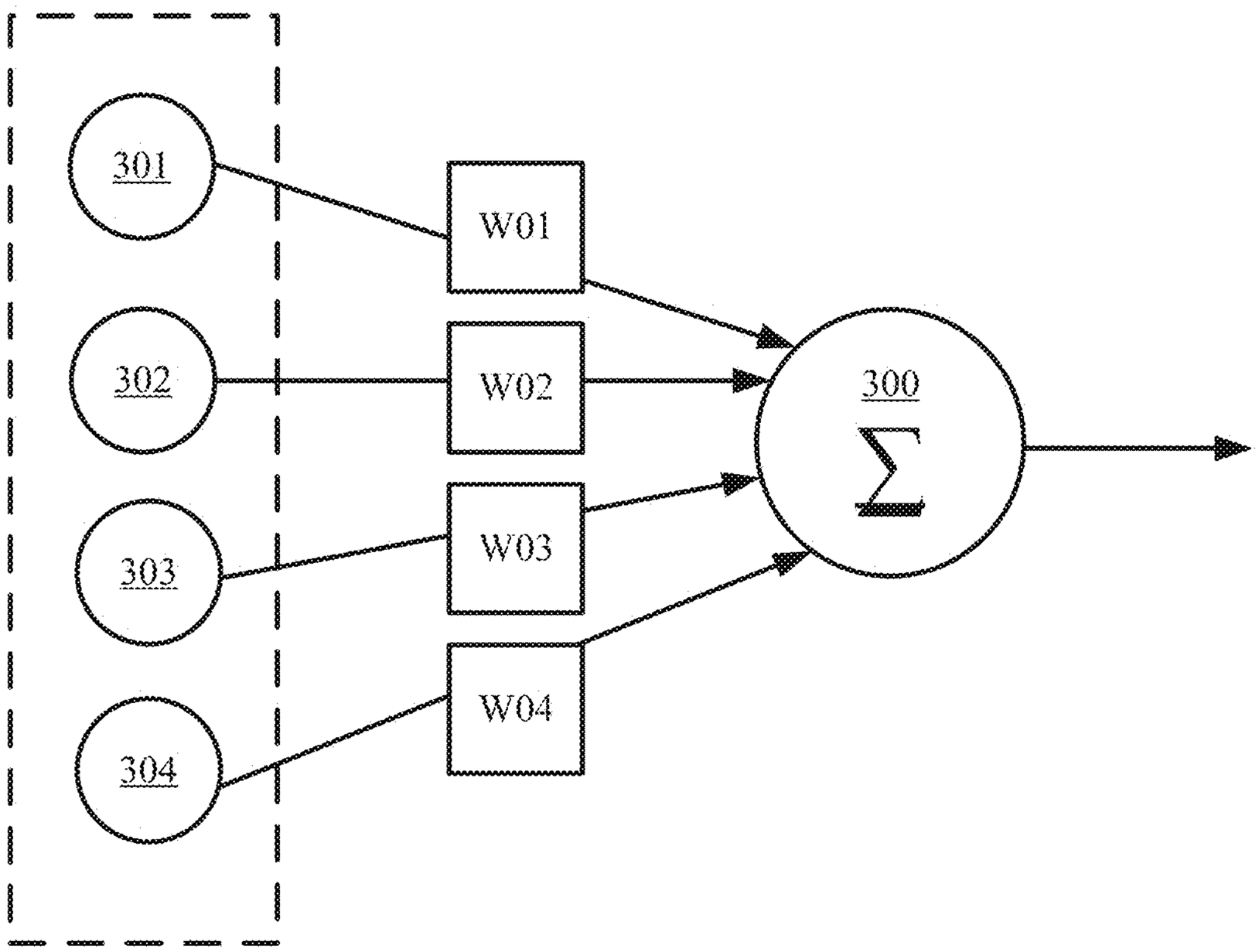


FIG. 3

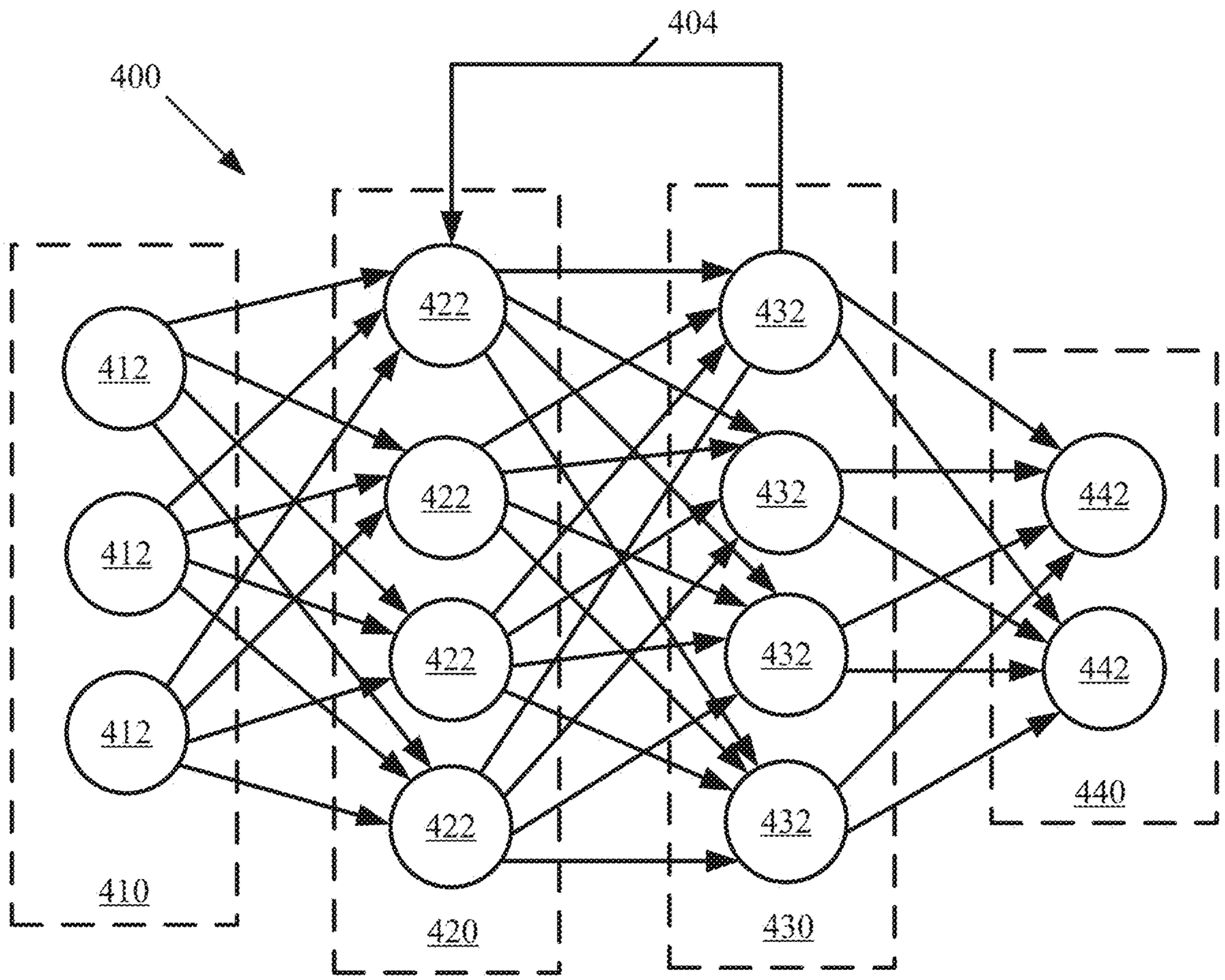

400
404
412
412
412
422
422
422
422
432
432
432
432
442
442
440
410
420
430

FIG. 4

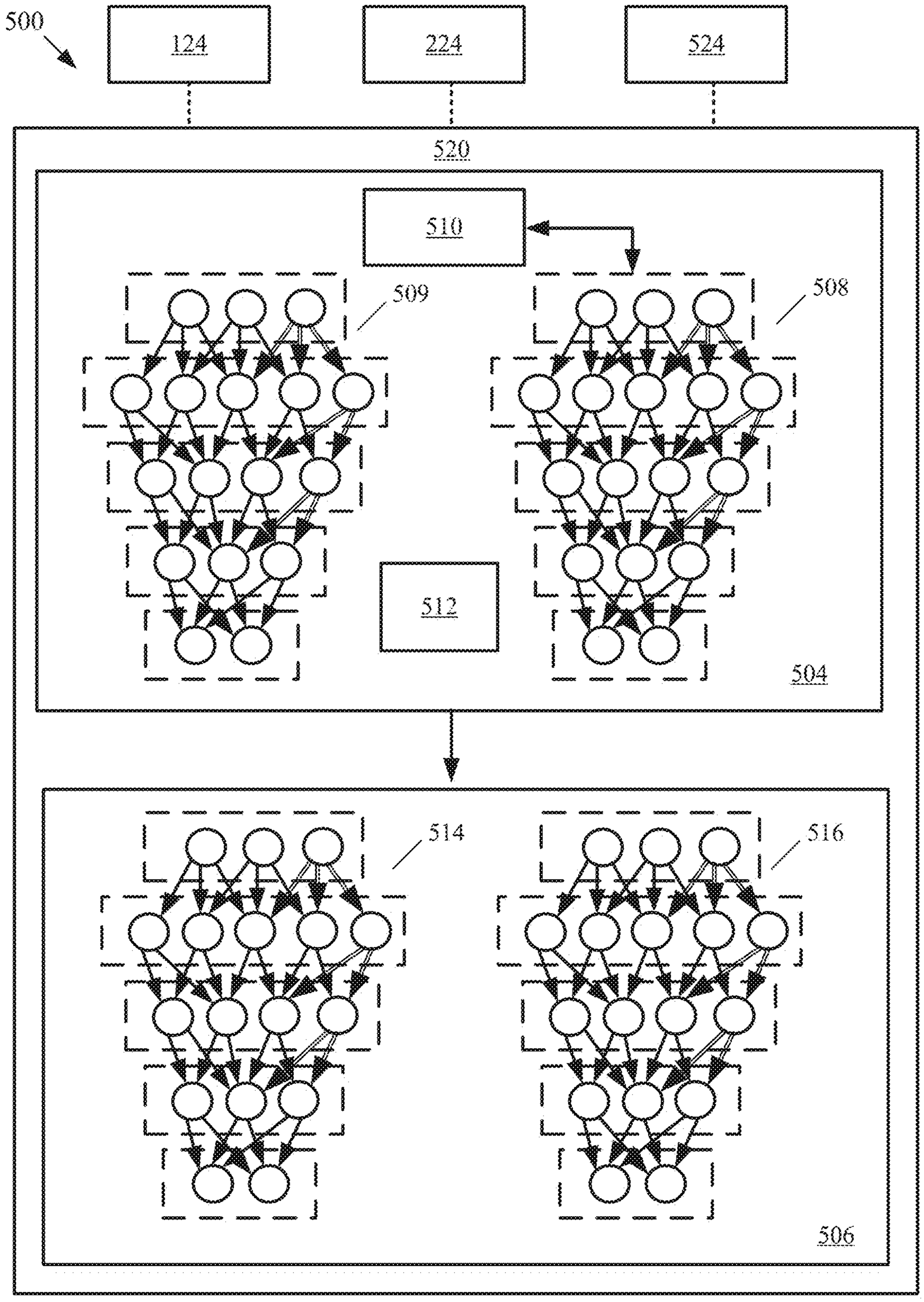


FIG. 5

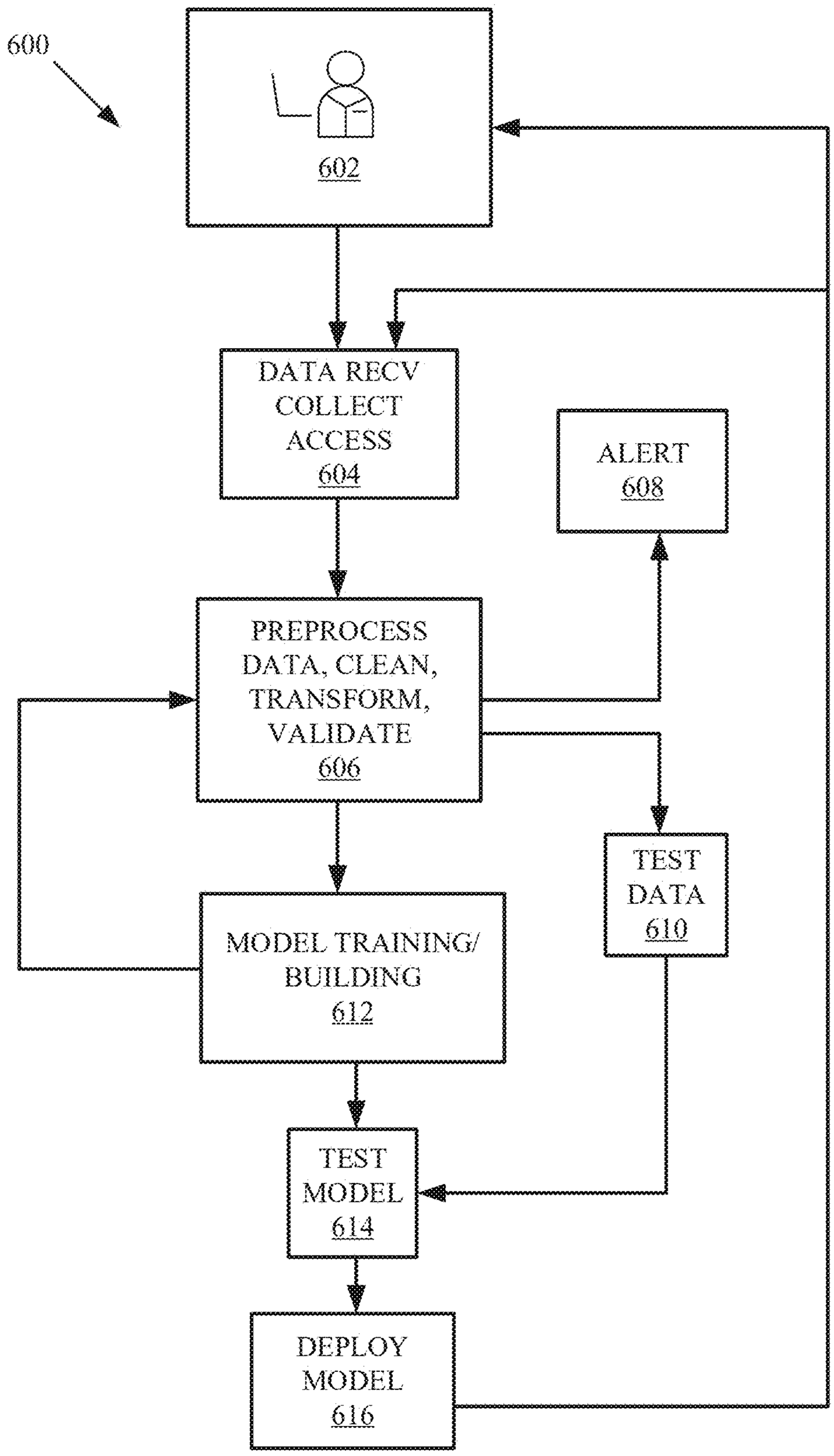

600
602
DATA RECV
COLLECT
ACCESS
604
ALERT
608
PREPROCESS
DATA, CLEAN,
TRANSFORM,
VALIDATE
606
TEST
DATA
610
MODEL TRAINING/
BUILDING
612
TEST
MODEL
614
DEPLOY
MODEL
616

FIG. 6

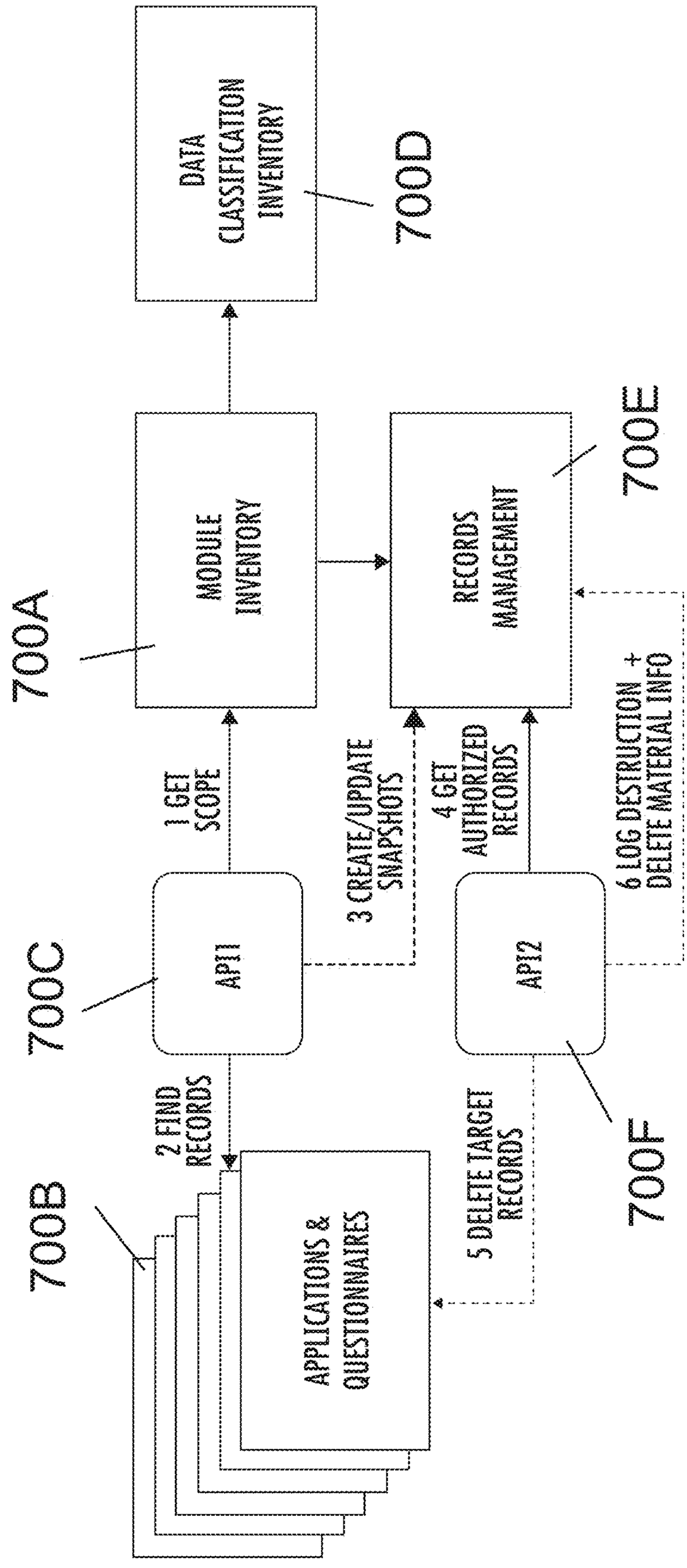
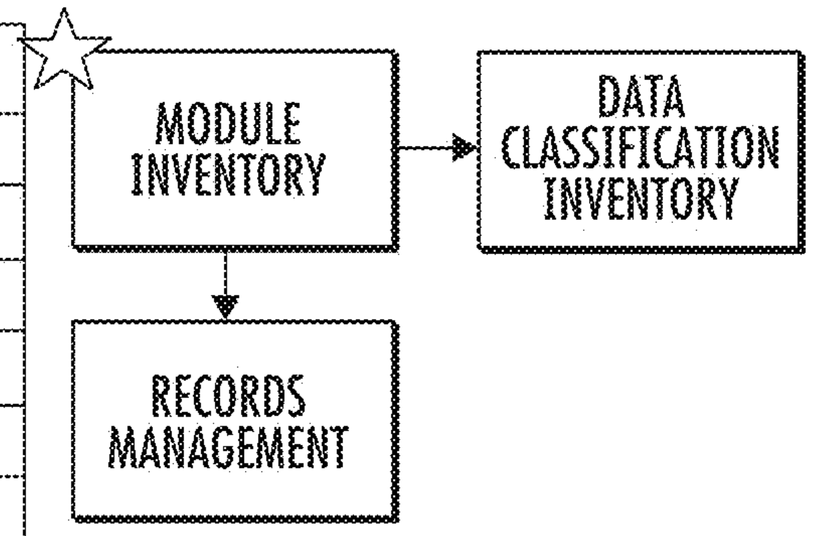

DATA CLASSIFICATION INVENTORY
700D
700A
MODULE INVENTORY
RECORDS MANAGEMENT
700E
700C
API1
API2
1 GET SCOPE
3 CREATE/UPDATE SNAPSHOTS
4 GET AUTHORIZED RECORDS
6 LOG DESTRUCTION + DELETE MATERIAL INFO
2 FIND RECORDS
5 DELETE TARGET RECORDS
700B
APPLICATIONS & QUESTIONNAIRES
700F

FIG. 7

800

MODULE INVENTORY	
MODULE NAME	TEXT (KEY)
MODULE STATUS	VALUES LIST
APPLICATION OWNER	RECORD PERMISSION
RETENTION LEAD	RECORD PERMISSION
DATE FIELD	TEXT
DATA CLASSIFICATION	CROSS REFERENCE
RETENTION PERIOD DAYS	NUMBER (CALCULATED)
MODULE ID	NUMBER
DATE FIELD ID	NUMBER



EDIT VIEW SAVE SAVE AND CLOSE RECORD SETUP

FIRST PUBLISHED: 5/13/2021 8:01 PM LAST UPDATED: 8/26/2024 2:46 PM

MODULE INFO METADATA DATA RETENTION & DESTRUCTION

▽ DATA RETENTION & DESTRUCTION

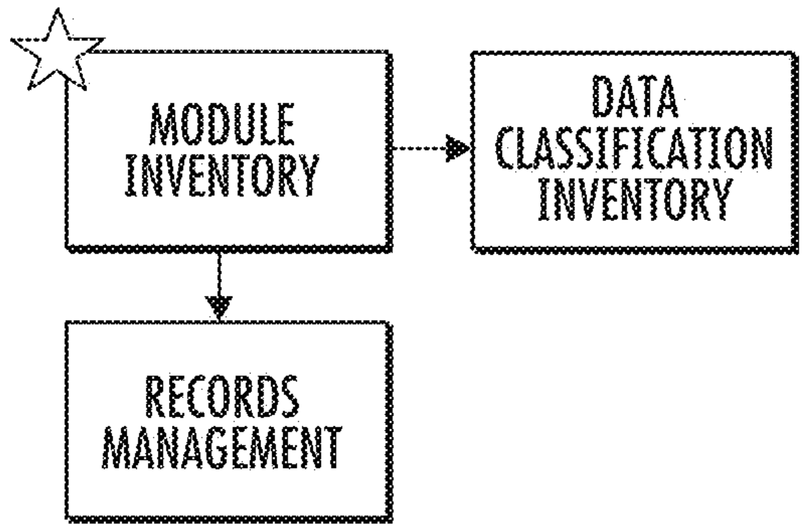
TO BEGIN THE ENROLLMENT PROCESS FOR THIS APPLICATION/QUESTIONNAIRE SELECT THE (...) ICON NEXT TO THE RETENTION PERIOD BELOW. SELECT THE CLASSIFICATION OF THE DATA FOUND IN THIS SPECIFIC MODULE.

RETENTION PERIOD: ... ENROLLMENT STATUS: NOT ENROLLED

*REQUIRED

FIG. 8

900



DATA FIELD AS ANCHOR FOR RETENTION

EDIT VIEW SAVE SAVE AND CLOSE

FIRST PUBLISHED: 5/13/2021 8:01 PM LAST UPDATED: 8/26/2024 2:46 PM

MODULE INFO | METADATA | DATA RETENTION & DESTRUCTION

▿ DATA RETENTION & DESTRUCTION

RETENTION PERIOD: BCH1000-BRANCH LOGS (1 YEAR) ⓧ ... ENROLLMENT STATUS: NOT ENROLLED

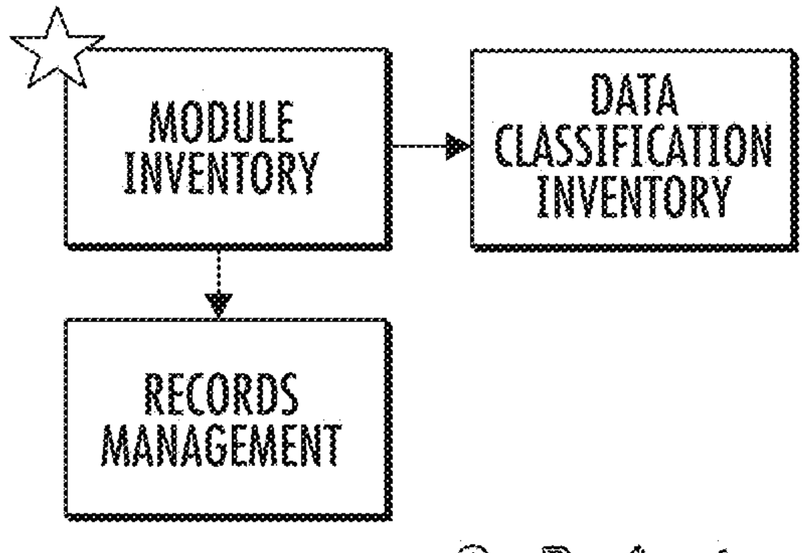
SELECT THE DATE FIELD IN THIS APPLICATON/QUESTIONNAIRE THAT SHOULD BE USED AS THE MARKER FOR DATA RETENTION.
IF THE RETENTION PERIOD IS "(INDEFINITE)" SELECT FIRST PUBLISHED.

* DATA FIELD TO BE USED FOR RETENTION PERIOD: ▿

*REQUIRED

FIG. 9

1000



RETENTION LEAD AND FIELDS TO DISPLAY

EDIT VIEW SAVE SAVE AND CLOSE

FIRST PUBLISHED: 5/13/2021 8:01 PM LAST UPDATED: 8/5/2024 8:00 PM

MODULE INFO | METADATA | DATA RETENTION & DESTRUCTION

▽ DATA RETENTION & DESTRUCTION

RETENTION PERIOD: BCH1000-BRANCH LOGS (1 YEAR) ⓧ ... ENROLLMENT STATUS: NOT ENROLLED

* DATA FIELD TO BE USED FOR RETENTION PERIOD: FIRST PUBLISHED ▽

SELECT THE PERSON(S) WHO WILL ACT AS THE DATA RETENTION LEAD FOR THE APPLICATION/QUESTIONNAIRE. THIS PERSON(S) WILL BE RESPONSIBLE FOR REVIEWING THE RECORDS OUTSIDE OF THE RETENTION PERIOD AND AUTHORIZING THEIR DESTRUCTION. ADDITIONALLY, THIS PERSON(S) WILL BE RESPONSIBLE FOR TAGGING RECORD AS "ON HOLD" FOR DATA RETENTION BEYOND THE RETENTION PERIOD. IF THE RETENTION PERIOD IS "(INDEFINITE)" SELECT YOUR NAME.

TYPE THE NAME OF UP TO THREE FIELDS FROM THE MODULE THAT YOU WOULD LIKE TO DISPLAY WITHIN THIS SOLUTION TO PROVIDE CONTEXT TO THE RECORDS BEING REVIEWED FOR RETENTION/DESTRUCTION.

* RETENTION LEAD: ...

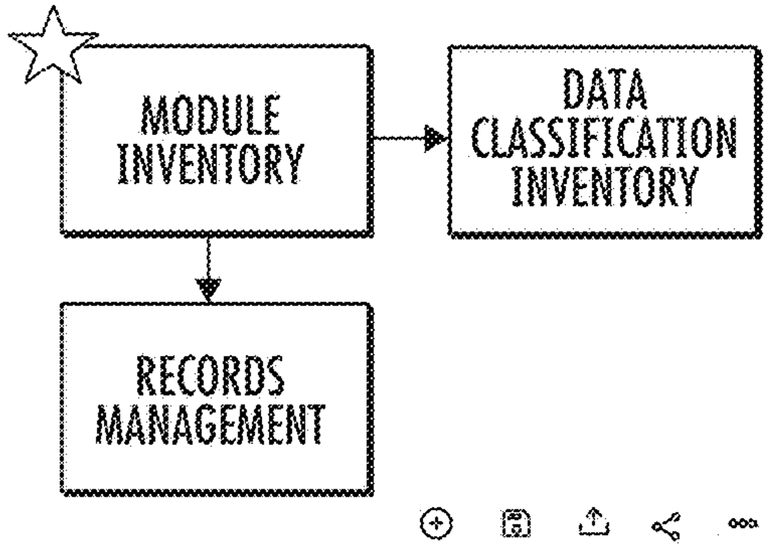
APPLICATION FIELD TO DISPLAY 1: ...

APPLICATION FIELD TO DISPLAY 2: ...

APPLICATION FIELD TO DISPLAY 3: ...

FIG. 10

1100



CONFIRMATION OF ENROLLMENT

EDIT | VIEW | SAVE | SAVE AND CLOSE

FIRST PUBLISHED: 5/13/2021 8:01 PM LAST UPDATED: 8/26/2024 2:39 PM

MODULE INFO | METADATA | DATA RETENTION & DESTRUCTION

▿ DATA RETENTION & DESTRUCTION

RETENTION PERIOD: BCH1000-BRANCH LOGS (1 YEAR) ⓧ ... ENROLLMENT STATUS: PENDING CONFIRMATION

* DATA FIELD TO BE USED FOR RETENTION PERIOD: FIRST PUBLISHED ▿

* RETENTION LEAD: VISION, SOLUTION OWNER ...

APPLICATION FIELD TO DISPLAY 1: BRANCH NAME ...

APPLICATION FIELD TO DISPLAY 2: BRANCH ID ...

APPLICATION FIELD TO DISPLAY 3: BRANCH MANAGER ...

CONFIRMATION STATEMENT: DATA WITHIN THE 'FACILITIES EMERGENCY RESPONSE PLANE' IS AUTHORIZED TO BE INCLUDED WITHIN THE DATA RETENTION AND DESTRUCTION, SOLUTION DATA IDENTIFIED WITHIN THE 1 YEAR (365 DAYS | CURRENTLY 8/27/2023) RETENTION PERIOD, BASED ON THE 'FIRST PUBLISHED' FIELD WILL BE COLLECTED AND PRESENTED WITHIN THIS TOOL FOR RETENTION OR DESTRUCTION CLASSIFICATION. UPON AUTHORIZATION BY THE RETENTION COORDINATOR ALL RECORDS CLASSIFIED FOR DESTRUCTION WILL BE PERMANENTLY DELETED OUT OF STORAGE AND RECOVERY OF THE DESTROYED RECORDS WILL NOT BE POSSIBLE. IF YOU AGREE TO THIS, CLICK THE CHECKBOX BELOW AND THEN CLICK THE 'SAVE' BUTTON.

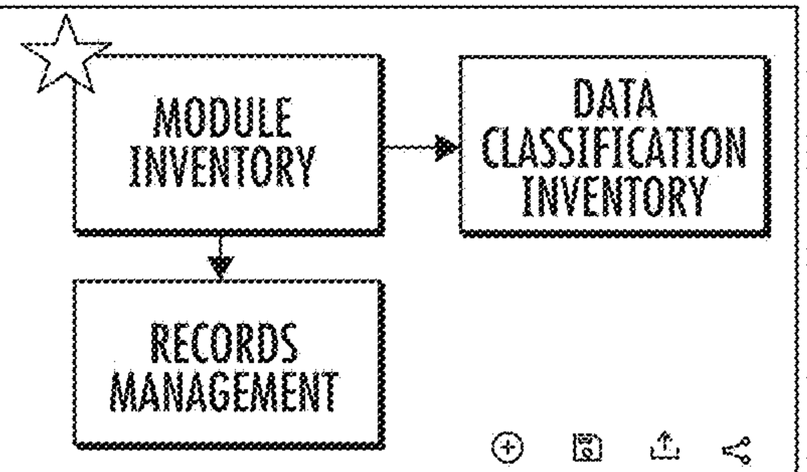
REVIEW CONFIRMATION: ☑ THE TEAM MAY PROCEED WITH CONFIGURING THE DATA RETENTION AND DESTRUCTION IMPLEMENTATION.

FIG. 11

1200

ADMINISTRATIVE SETUP

EDIT VIEW SAVE SAVE AND CLOSE



FIRST PUBLISHED: 5/13/2021 8:01 PM LAST UPDATED: 8/26/2024 2:55 PM

MODULE INFO | METADATA | DATA RETENTION & DESTRUCTION

▿ DATA RETENTION & DESTRUCTION

RETENTION PERIOD: BCH1000-BRANCH LOGS (1 YEAR)

ENROLLMENT STATUS: ENROLLED - PENDING CONFIGURATION

DATA FIELD TO BE USED FOR RETENTION PERIOD: FIRST PUBLISHED
PERIOD:

RETENTION LEAD: VISION, SOLUTION OWNER

APPLICATION FIELD TO DISPLAY 1: BRANCH NAME
APPLICATION FIELD TO DISPLAY 2: BRANCH ID
APPLICATION FIELD TO DISPLAY 3: BRANCH MANAGER

REVIEW CONFIRMATION: THE TEAM MAY PROCEED WITH CONFIGURING THE DATA RETENTION AND DESTRUCTION IMPLEMENTATION.

▿ DATA RETENTION & DESTRUCTION (CONFIGURATION)

TRACKING ID FIELD ID:	111111
FIRST PUBLISHED FIELD ID:	111112
LAST UPDATED FIELD ID:	111113
DATE FIELD ID:	111114
FIELD ID 1:	111115
FIELD ID 2:	111116
FIELD ID 3:	111117

DATE FIELD NAME: FIRST PUBLISHED
FIELD 1 NAME: BRANCH NAME
FIELD 2 NAME: BRANCH ID
FIELD 3 NAME: BRANCH MANAGER
UNENROLL MODULE: ☐ YES, CLEAR ALL CONFIRMATION AND UNENROLL THE MODULE

CONFIGURATION CONFIRMATION: ☑ THE ABOVE FIELD IDS ARE THE IDS OF THE FIELDS SPECIFIED BY THE 'RETENTION LEAD' FOR THIS ENVIRONMENT

FIG. 12

1300

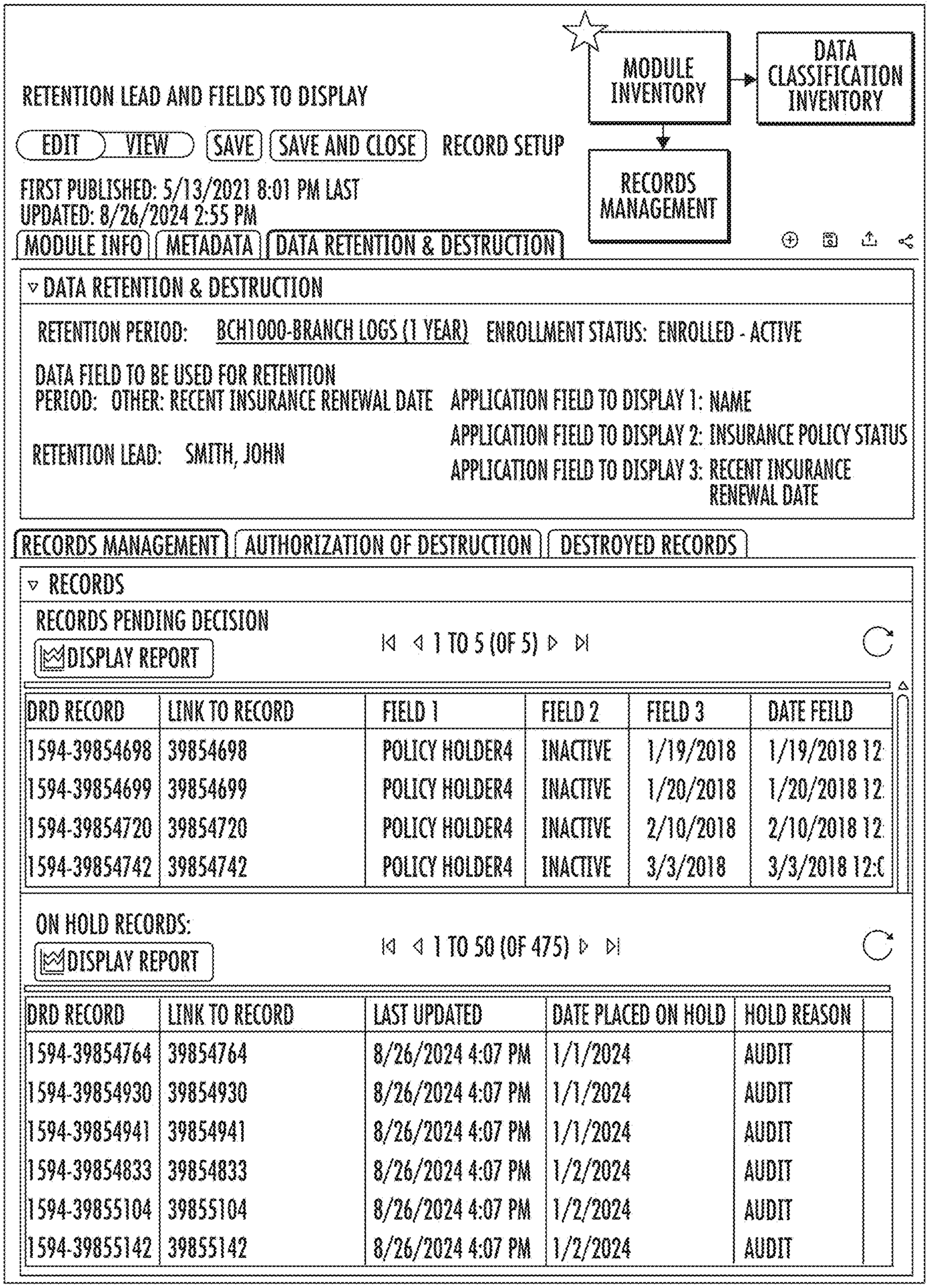

MODULE INVENTORY
DATA CLASSIFICATION INVENTORY
RECORDS MANAGEMENT
RETENTION LEAD AND FIELDS TO DISPLAY
EDIT VIEW SAVE SAVE AND CLOSE RECORD SETUP
FIRST PUBLISHED: 5/13/2021 8:01 PM LAST UPDATED: 8/26/2024 2:55 PM
MODULE INFO METADATA DATA RETENTION & DESTRUCTION
DATA RETENTION & DESTRUCTION
RETENTION PERIOD: BCH1000-BRANCH LOGS (1 YEAR) ENROLLMENT STATUS: ENROLLED - ACTIVE
DATA FIELD TO BE USED FOR RETENTION PERIOD: OTHER: RECENT INSURANCE RENEWAL DATE
RETENTION LEAD: SMITH, JOHN
APPLICATION FIELD TO DISPLAY 1: NAME
APPLICATION FIELD TO DISPLAY 2: INSURANCE POLICY STATUS
APPLICATION FIELD TO DISPLAY 3: RECENT INSURANCE RENEWAL DATE
RECORDS MANAGEMENT AUTHORIZATION OF DESTRUCTION DESTROYED RECORDS
RECORDS
RECORDS PENDING DECISION
DISPLAY REPORT
1 TO 5 (OF 5)
DRD RECORD | LINK TO RECORD | FIELD 1 | FIELD 2 | FIELD 3 | DATE FEILD
1594-39854698 | 39854698 | POLICY HOLDER4 | INACTIVE | 1/19/2018 | 1/19/2018 12
1594-39854699 | 39854699 | POLICY HOLDER4 | INACTIVE | 1/20/2018 | 1/20/2018 12
1594-39854720 | 39854720 | POLICY HOLDER4 | INACTIVE | 2/10/2018 | 2/10/2018 12
1594-39854742 | 39854742 | POLICY HOLDER4 | INACTIVE | 3/3/2018 | 3/3/2018 12:(
ON HOLD RECORDS:
DISPLAY REPORT
1 TO 50 (OF 475)
DRD RECORD | LINK TO RECORD | LAST UPDATED | DATE PLACED ON HOLD | HOLD REASON
1594-39854764 | 39854764 | 8/26/2024 4:07 PM | 1/1/2024 | AUDIT
1594-39854930 | 39854930 | 8/26/2024 4:07 PM | 1/1/2024 | AUDIT
1594-39854941 | 39854941 | 8/26/2024 4:07 PM | 1/1/2024 | AUDIT
1594-39854833 | 39854833 | 8/26/2024 4:07 PM | 1/2/2024 | AUDIT
1594-39855104 | 39855104 | 8/26/2024 4:07 PM | 1/2/2024 | AUDIT
1594-39855142 | 39855142 | 8/26/2024 4:07 PM | 1/2/2024 | AUDIT

FIG. 13

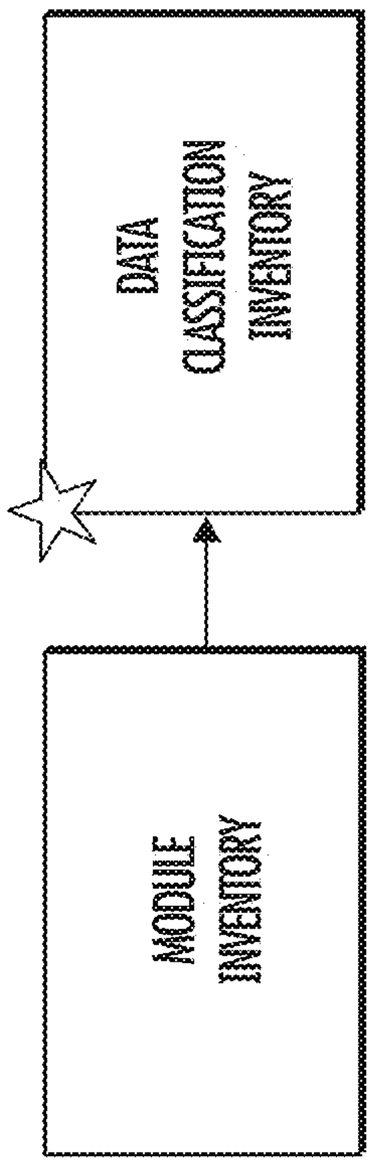


FIG. 14

1400

DATA CLASSIFICATION INVENTORY	
CLASSIFICATION	TEXT (KEY)
CATEGORY	VALUES LIST
RETENTION PERIOD	TEXT
RETENTION PERIOD DAYS	NUMBER

1500

SELECTING DATA CLASSIFICATION

MODULE INVENTORY → DATA CLASSIFICATION INVENTORY

RECORDS MANAGEMENT

RECORD LOOKUP

RETENTION PERIOD FILTER CRITERIA

SHOW FILTERS

SEARCH: FIND

	RETENTION SERIES CATEGORY NAME 1	NAME	DESCRIPTION	DEFINING REGULATION	INFORMATION OWNER	RETENTION PERIOD	RETENTION PERIOD IN DAYS	RETENTION SERIES CODE 2	RETENTI SERIES DISCRIP
⊞	RETENTION SERIES CATEGORY NAME: AUDITING								
⊞	RETENTION SERIES CATEGORY NAME: BANK - BRANCH OPERATIONS								
⊞	RETENTION SERIES CATEGORY NAME: CREDIT AND LENDING								
⊞	RETENTION SERIES CATEGORY NAME: ELECTRONIC COMMUNICATIONS								
⊟	RETENTION SERIES CATEGORY NAME: GENERAL								
○	GENERAL	POLICIES - REGULATED	REGULATED INFORMATION OF ORGANIZATION -APPROVED METHODS OR PROCESSES FOR PERFORMING ACTIVITIES TO ENSURE UNIFORMITY AND COMPLIANCE WITH INSTITUTION AND LEAGAL REQUIREMENTS (E.G., SEC).	17 C.F.R. 270.17E-1(D)(1)		INDEFINITE	999999	GEN1000	DATA DO ORGANI PLANNI IN ALL A INCLUDI
○	GENERAL	POLICIES - NON-REGULATED	NON-REGULATED POLICIES AND INFORMATION OF ORGANIZATION-APPROVED METHODS OR PROCESSES FOR PERFORMING ACTIVITIES TO ENSURE UNIFORMITY AND COMPLIANCE WITH INSTITUTION AND LEGAL REQUIREMENTS. SEE GEN2010 FOR PROCEDURES AND MANUALS - NON-REGULATED AND GEN1000 FOR REGULATED POLICIES.	PA. INS. BULLETIN NO. 2011-10, 41 PA.B. 5849(27)		SUPERSEDED + 7 YEARS	2556	GEN1010	DATA DO ORGANI AND MA IN ALL A ARE NOT RETENTI SCHEDU
○	GENERAL	REGULATED PROCEDURES & MANUALS	PROCEDURES AND MANUALS OF REGULATED INFORMATION OF ORGANIZATION-APPROVED METHODS OR PROCESSES FOR PERFORMING ACTIVITES TO	80 F.R. 14437 RULE 13N-7(B)		SUPERSEDED + 7 YEARS	2556	GEN2000	DATA DO ORGANI ADMINI

FIG. 15

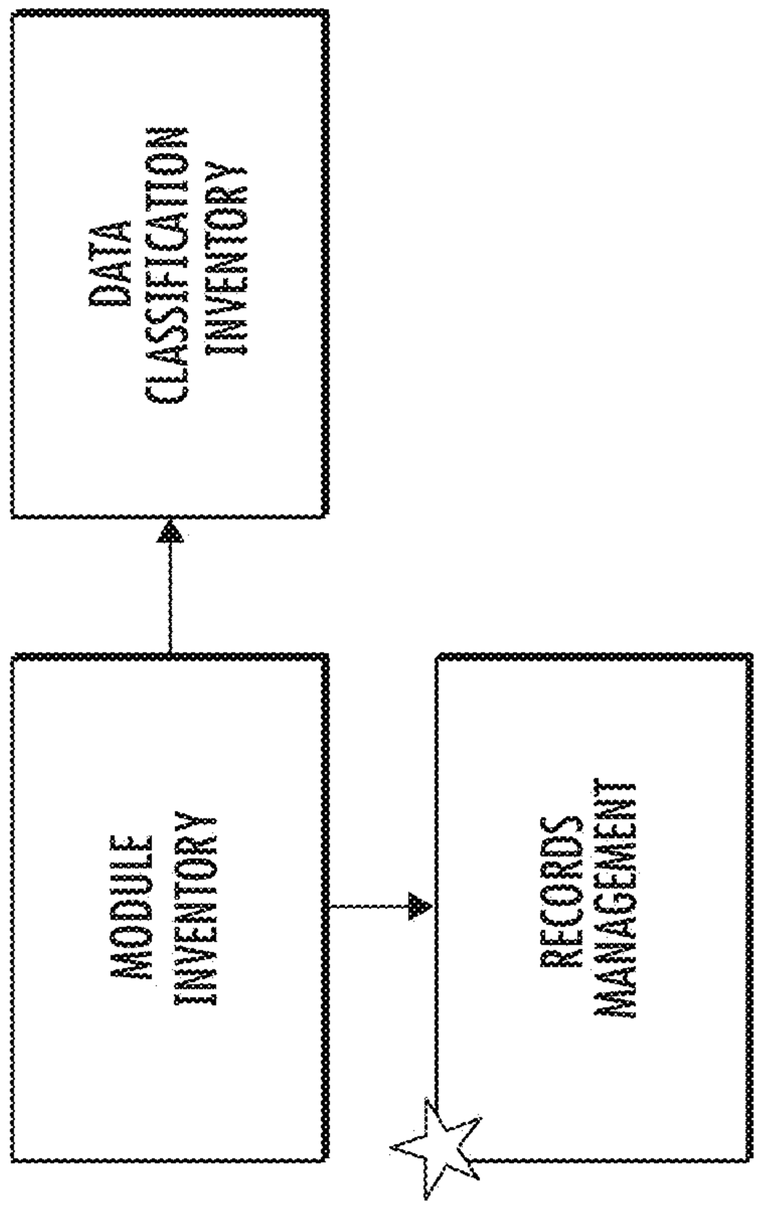
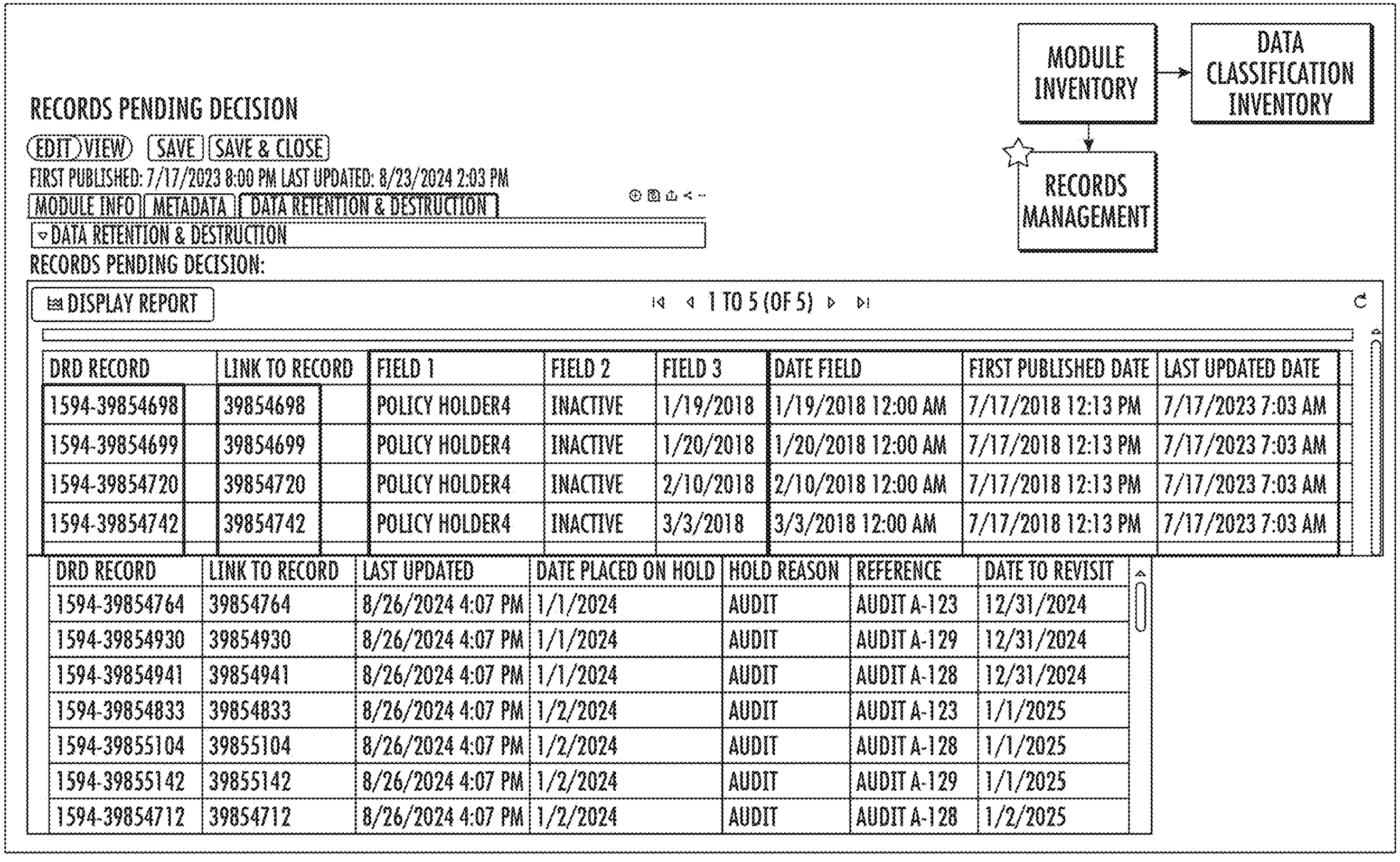


FIG. 16

1600

RECORDS MANAGEMENT	
RECORD ID	TEXT (CALCULATED)
MODULE NAME	TEXT (CALCULATED)
DATE FIELD VALUE	DATE
HOLD	VALUES LIST (YES/NO)
HOLD REASON	VALUES LIST (AUDIT/LEGAL/ETC)
DATE PLACED ON HOLD	DATE
DATE TO REVISIT	DATE
MODULE ID	NUMBER
CONTENT ID	NUMBER

1700



DRD RECORD	LINK TO RECORD	FIELD 1	FIELD 2	FIELD 3	DATE FIELD	FIRST PUBLISHED DATE	LAST UPDATED DATE
1594-39854698	39854698	POLICY HOLDER4	INACTIVE	1/19/2018	1/19/2018 12:00 AM	7/17/2018 12:13 PM	7/17/2023 7:03 AM
1594-39854699	39854699	POLICY HOLDER4	INACTIVE	1/20/2018	1/20/2018 12:00 AM	7/17/2018 12:13 PM	7/17/2023 7:03 AM
1594-39854720	39854720	POLICY HOLDER4	INACTIVE	2/10/2018	2/10/2018 12:00 AM	7/17/2018 12:13 PM	7/17/2023 7:03 AM
1594-39854742	39854742	POLICY HOLDER4	INACTIVE	3/3/2018	3/3/2018 12:00 AM	7/17/2018 12:13 PM	7/17/2023 7:03 AM

DRD RECORD	LINK TO RECORD	LAST UPDATED	DATE PLACED ON HOLD	HOLD REASON	REFERENCE	DATE TO REVISIT
1594-39854764	39854764	8/26/2024 4:07 PM	1/1/2024	AUDIT	AUDIT A-123	12/31/2024
1594-39854930	39854930	8/26/2024 4:07 PM	1/1/2024	AUDIT	AUDIT A-129	12/31/2024
1594-39854941	39854941	8/26/2024 4:07 PM	1/1/2024	AUDIT	AUDIT A-128	12/31/2024
1594-39854833	39854833	8/26/2024 4:07 PM	1/2/2024	AUDIT	AUDIT A-123	1/1/2025
1594-39855104	39855104	8/26/2024 4:07 PM	1/2/2024	AUDIT	AUDIT A-128	1/1/2025
1594-39855142	39855142	8/26/2024 4:07 PM	1/2/2024	AUDIT	AUDIT A-129	1/1/2025
1594-39854712	39854712	8/26/2024 4:07 PM	1/2/2024	AUDIT	AUDIT A-128	1/2/2025

FIG. 17

1800

(PLACING A RECORD ON HOLD)

EDIT VIEW SAVE SAVE & CLOSE

▿ MODULE INFORMATION

MODULE ID	MODULE NAME	DATE FIELD TO BE USED FOR RETENTION PERIOD	RETENTION PERIOD	RETENTION LEAD
1,594	DRD TEST APP1	OTHER: RECENT INSURANCE RENEWAL DATE	BCH1000 - BRANCH LOGS (1 YEAR)	DOE, JOHN

▿ DATA RETENTION & DESTRUCTION

DRD ID: 1594-39854698 STATUS: AUTHORIZED FOR DESTRUCTION
AUTHORIZATION DATE: 8/27/2024

▿ RECORD INFORMATION

MODULE NAME:	DRD_TEST_APP1	LINK TO RECORD:	39854698
FIELD 1:	POLICYHOLDER4	DATE FIELD:	1/19/2023 12:00 AM
FIELD 2:	INACTIVE	FIRST PUBLISHED DATE:	7/17/2023 12:13 PM
FIELD 3:	1/19/2018	LAST UPDATED DATE:	7/17/2023 7:03 AM

▿ HOLD DETAILS

HOLD RECORD: ◉ YES ○ NO

DATE PLACED ON HOLD: 8/28/2024

*HOLD REASON: ○ AUDIT ○ LEGAL ○ OTHER ○ COMPLIANCE ○ MANAGER

*HOLD JUSTIFICATION:

*DATE TO REVISIT:

REFERENCE:

FIG. 18

1900

VIEWING "ON HOLD" RECORDS

EDIT VIEW | SAVE | SAVE & CLOSE

FIRST PUBLISHED: 7/17/2023 8:00 PM LAST UPDATED: 8/23/2024 2:03 PM

MODULE INFO | METADATA | DATA RETENTION & DESTRUCTION

DATA RETENTION & DESTRUCTION

RETENTION PERIOD:	BCH1000 - BRANCH LOGS (1 YEAR)	ENROLLMENT STATUS:	ENROLLED - ACTIVE
DATE FIELD TO BE USED FOR RETENTION PERIOD:	OTHER: RECENT INSURANCE RENEWAL DATE		
RETENTION LEAD:	SMITH, JOHN	APPLICATION FIELD TO DISPLAY 1:	NAME
		APPLICATION FIELD TO DISPLAY 2:	INSURANCE POLICY STATUS
		APPLICATION FIELD TO DISPLAY 3:	RECENT INSURANCE RENEWAL DATE

RECORDS MANAGEMENT | AUTHORIZATION OF DESTRUCTION | DESTROYED RECORDS

RECORDS

RECORDS PENDING DECISION

ON HOLD RECORDS:

DISPLAY REPORT 1 TO 50 (OF 475)

DRD RECORD	LINK TO RECORD	LAST UPDATED	DATE PLACED ON HOLD	HOLD REASON	REFERENCE	DATE TO REVISIT
1594-39854764	39854764	8/26/2024 4:07 PM	1/1/2024	AUDIT	AUDIT A-123	12/31/2024
1594-39854930	39854930	8/26/2024 4:07 PM	1/1/2024	AUDIT	AUDIT A-129	12/31/2024
1594-39854941	39854941	8/26/2024 4:07 PM	1/1/2024	AUDIT	AUDIT A-128	12/31/2024
1594-39854833	39854833	8/26/2024 4:07 PM	1/2/2024	AUDIT	AUDIT A-123	1/1/2025
1594-39855104	39855104	8/26/2024 4:07 PM	1/2/2024	AUDIT	AUDIT A-128	1/1/2025
1594-39855142	39855142	8/26/2024 4:07 PM	1/2/2024	AUDIT	AUDIT A-129	1/1/2025
1594-39854712	39854712	8/26/2024 4:07 PM	1/2/2024	AUDIT	AUDIT A-128	1/2/2025

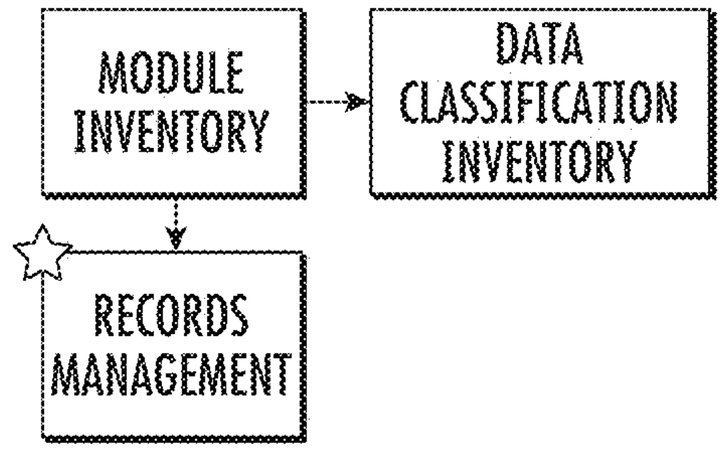


FIG. 19

2000

AUTHORIZE DESTRUCTION

EDIT VIEW | SAVE | SAVE & CLOSE

FIRST PUBLISHED: 7/17/2023 8:00 PM LAST UPDATED: 8/23/2024 2:03 PM

MODULE INFO | METADATA | DATA RETENTION & DESTRUCTION

DATA RETENTION & DESTRUCTION

RETENTION PERIOD:	BCH1000 - BRANCH LOGS (1 YEAR)	ENROLLMENT STATUS:	ENROLLED - ACTIVE
DATE FIELD TO BE USED FOR RETENTION PERIOD:	OTHER: RECENT INSURANCE RENEWAL DATE		
RETENTION LEAD:	SMITH, JOHN	APPLICATION FIELD TO DISPLAY 1:	NAME
		APPLICATION FIELD TO DISPLAY 2:	INSURANCE POLICY STATUS
		APPLICATION FIELD TO DISPLAY 3:	RECENT INSURANCE RENEWAL DATE

RECORDS MANAGEMENT | AUTHORIZATION OF DESTRUCTION | DESTROYED RECORDS

AUTHORIZATION OF DESTRUCTION

AUTHORIZE: ☐ I AUTHORIZE AUTOMATICALLY DELETING THE RECORDS LISTED IN THE BELOW REPORT

RECORDS TO BE AUTHORIZED FOR DESTRUCTION:

DISPLAY REPORT 1 TO 5 (OF 5)

DRD RECORD	LINK TO RECORD	FIELD 1	FIELD 2	FIELD 3	DATE FIELD	FIRST PUBLISHED DATE	LAST UPDATED DATE
1594-39854931	39854931	POLICY HOLDER4	INACTIVE	9/8/2018		7/17/2023 12:13 PM	7/17/2023 7:03 AM
1594-39854940	39854940	POLICY HOLDER4	INACTIVE	9/17/2018		7/17/2023 12:13 PM	7/17/2023 7:03 AM
1594-39854942	39854942	POLICY HOLDER4	INACTIVE	9/19/2018		7/17/2023 12:13 PM	7/17/2023 7:03 AM
1594-39854943	39854943	POLICY HOLDER4	INACTIVE	9/20/2018		7/17/2023 12:13 PM	7/17/2023 7:03 AM

FIG. 20

2100

DESTROYED RECORDS - THE LOG

EDIT VIEW | SAVE | SAVE & CLOSE

FIRST PUBLISHED: 11/8/2016 8:00 PM LAST UPDATED: 8/27/2024 8:00 PM

MODULE INFO | METADATA | DATA RETENTION & DESTRUCTION

DATA RETENTION & DESTRUCTION

RETENTION PERIOD:	BCH1000 - BRANCH LOGS (1 YEAR)	ENROLLMENT STATUS:	ENROLLED - ACTIVE
DATE FIELD TO BE USED FOR RETENTION PERIOD:	FIRST PUBLISHED		
RETENTION LEAD:	SMITH, JOHN	APPLICATION FIELD TO DISPLAY 1:	APPLICATION ID
		APPLICATION FIELD TO DISPLAY 2:	APPLICATION NAME
		APPLICATION FIELD TO DISPLAY 3:	PRODUCTION STATUS

RECORDS MANAGEMENT | AUTHORIZATION OF DESTRUCTION | DESTROYED RECORDS

DESTROYED RECORDS

DESTROYED RECORDS:

DISPLAY REPORT 1 TO 15 (OF 15)

DRD RECORD	LINK TO RECORD	AUTHORIZATION DATE	DESTRUCTION DATE/TIME	
70-11677607	11677607	7/14/2023	7/14/2023 1:43 PM	
70-11677608	11677608	7/14/2023	7/14/2023 1:44 PM	
70-11677609	11677609	7/14/2023	7/14/2023 1:44 PM	
70-11677610	11677610	7/14/2023	7/14/2023 1:44 PM	
70-12659704	12659704	7/14/2023	7/14/2023 1:44 PM	
70-12662394	12662394	7/14/2023	7/14/2023 1:44 PM	
70-12662395	12662395	7/14/2023	7/14/2023 1:44 PM	

FIG. 21

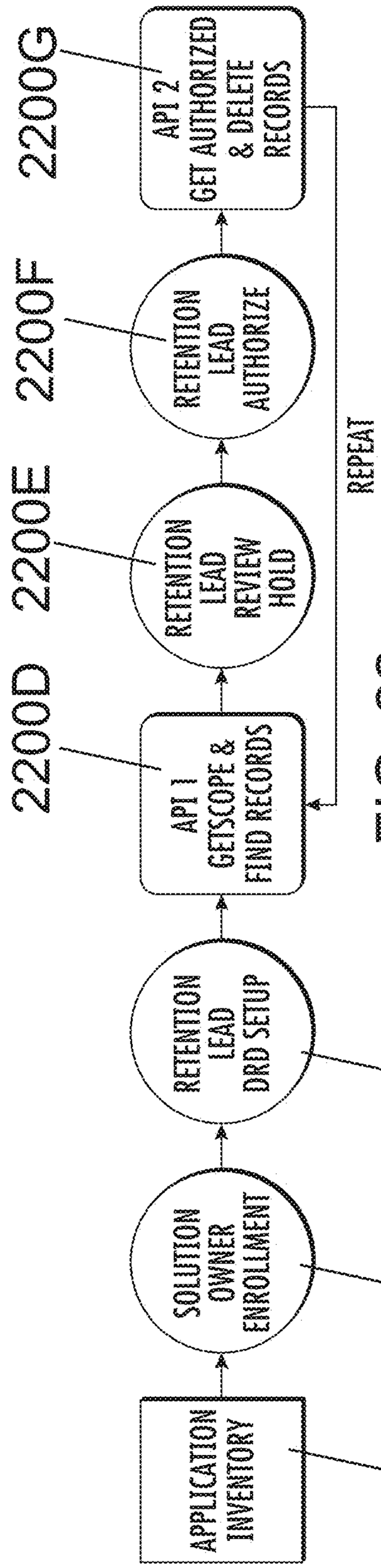

APPLICATION INVENTORY
SOLUTION OWNER ENROLLMENT
RETENTION LEAD DRD SETUP
API 1 GETSCOPE & FIND RECORDS
RETENTION LEAD REVIEW HOLD
RETENTION LEAD AUTHORIZE
API 2 GET AUTHORIZED & DELETE RECORDS
REPEAT
2200A
2200B
2200C
2200D
2200E
2200F
2200G

FIG. 22

2300

DATA FEED - LOCK AUTHORIZED RECORDS

TAG SNAPSHOTS WITH AUTHORIZATION DATE AND API GREEN LIGHT

GENERAL | SOURCE CONNECTION | SOURCE PARSING | SOURCE DEFINITION | DATA MAP | RUN CONFIGURAION

FIELD MAP | KEY FIELD DEFINITION

SOURCE FIELDS

DATA RETENTION AND DESTRUCTION REPOSITORY

TARGET FIELD	FIELD TYPE	SOURCE FIELD ↓	OPTIONS	ACTIONS
FILTER	FILTER	FILTER		
MODULE INFORMATION	CROSS REFERENCE	FIELD_CONTENTID		
DRO ID	TEXT	DRD_ID		
AUTHORIZATION DATE	DATE	AUTH_DATE		
▹ OLH: API STATUS	VALUES LIST	RECORD		

FIG. 23

2400

QUICK LINKS ADD A NEW ISSUE ADD NEW BAU REQUEST

DATA RETENTION AND DESTRUCTION

SOLUTION ENROLLMENT REPORT

SOLUTION NAME 1	MODULE NAME 2	ENROLLMENT STATUS
APPLICATION ASSESSMENT	ACTION ITEMS	PENDING CONFIRMENTION
APPLICATION PORTFOLIO MANAGEMENT (APM)	APPLICATIONS	PENDING CONFIRMENTION
AUDIT	APPOINTMENT	NOT ENROLLED
	AUDIT ATTACHMENTS	NOT ENROLLED
	AUDIT BUSINESS PROCESSES	NOT ENROLLED
	AUDIT CONTROL SELF ASSESSMENTS	NOT ENROLLED

ENROLLED APPLICATIONS

5

NOT ENROLLED APPLICATIONS

454

MY RECORDS PENDING DECISION

MODULE NAME

ZDRD - TEST MODULE 1000

DRD RECORD	FIELD 1	FIELD 2	FIELD 3	DATE FIELD	FIRST PUBLISHED DATE	LAST UPDATED DATE	LINK TO RECORD
1639-59392932	A TEST	THIS	IS	9/14/2014 8:00 PM	1/11/2024 5:04 AM	1/11/2024 5:04 AM	59392932
1639-59392933	THIS	IS	A	8/30/2014 8:00 PM	1/11/2024 5:04 AM	1/11/2024 5:04 AM	59392933
1639-59392934	IS	A	TEST	8/15/2014 8:00 PM	1/11/2024 5:04 AM	1/11/2024 5:04 AM	59392934
1639-59392935	A	TEST	JUST	7/31/2014 8:00 PM	1/11/2024 5:04 AM	1/11/2024 5:04 AM	59392935

PENDING CONFIRMATION

10

PENDING CONFIG

6

AUTHORIZED FOR DESTRUCTION (ALL MODULES)

4.1K

MY RECORDS ON HOLD

MODULE NAME

APPLICATIONS

DRD RECORD	DATE TO REVISIT	HOLD REASON	REFERENCE	FIELD 1	FIELD 2	FIELD 3	DATE FIELD	FIRST PUBLISHED DATE	LAST PUBLISHED DATE	LINK TO RECORD
70-7294083	7/13/2023	AUDIT	AUDIT A-129	APPID-10039569	NETWORK ACCESS PROVISIONING SYSTEM (NAPS)-ENTERPRISE	IN PRODUCTION	7/21/2022 5:19 AM	7/21/2022 5:19 AM	8/27/2024 9:16 AM	7294083

PAGE 1 OF 1 (1 RECORDS)

PAGE 1 OF 1 (1 RECORDS)

RECORDS ON HOLD (ALL MODULES)

478

FIG. 24

2500

INVENTORY: DRD_TEST_APP1

EDIT VIEW

FIRST PUBLISHED: 7/17/2023 8:00 PM LAST UPDATED: 9/20/2024 6:32 PM

MODULE INFO | METADATA | DATA RETENTION & DESTRUCTION

DATA RETENTION & DESTRUCTION

RETENTION PERIOD:	BCH1000 - BRANCH LOGS (1 YEAR)	ENROLLMENT STATUS:	ENROLLED - ACTIVE
DATE FIELD TO BE USED FOR RETENTION PERIOD:	OTHER: RECENT INSURANCE RENEWAL DATE		
RETENTION LEAD:	SMITH, JOHN	APPLICATION FIELD TO DISPLAY 1:	NAME
		APPLICATION FIELD TO DISPLAY 2:	INSURANCE POLICY STATUS
		APPLICATION FIELD TO DISPLAY 3:	RECENT INSURANCE RENEWAL DATE

RECORDS MANAGEMENT | AUTHORIZATION OF DESTRUCTION | DESTROYED RECORDS

RECORDS

RECORDS PENDING DECISION:

DISPLAY REPORT 1 TO 50 (OF 16,745)

DRD RECORD	LINK TO RECORD	FIELD 1	FIELD 2	FIELD 3	DATE FIELD	FIRST PUBLISHED DATE	LAST UPDATED DATE
1594-39837124	39837124	POLICY HOLDER1	ACTIVE	3/15/2022	3/14/2022 8:00 PM	7/17/2023 7:03 AM	7/17/2023 7:25 AM
1594-39837126	39837126	POLICY HOLDER1	ACTIVE	3/15/2022	3/14/2022 8:00 PM	7/17/2023 7:25 AM	7/17/2023 7:25 AM
1594-39837127	39837127	POLICY HOLDER1	INACTIVE	3/15/2022	3/14/2022 8:00 PM	7/17/2023 7:25 AM	7/17/2023 7:25 AM
1594-39837128	39837128	POLICY HOLDER1	INACTIVE	3/15/2022	3/14/2022 8:00 PM	7/17/2023 7:25 AM	7/17/2023 7:25 AM

ON HOLD RECORDS:

DISPLAY REPORT 1 TO 50 (OF 475)

DRD RECORD	LINK TO RECORD	LAST UPDATED	DATE PLACED ON HOLD	HOLD REASON	REFERENCE	DATE TO REVISIT
1594-39854764	39854764	9/17/2024 3:15 PM	1/1/2024	AUDIT	AUDIT A-123	12/31/2024
1594-39854930	39854930	9/17/2024 3:17 PM	1/1/2024	AUDIT	AUDIT A-129	12/31/2024
1594-39854941	39854941	9/17/2024 3:17 PM	1/1/2024	AUDIT	AUDIT A-128	12/31/2024
1594-39854833	39854833	9/17/2024 3:16 PM	1/2/2024	AUDIT	AUDIT A-123	1/1/2025
1594-39855104	39855104	9/17/2024 3:19 PM	1/2/2024	AUDIT	AUDIT A-128	1/1/2025
1594-39855142	39855142	9/17/2024 3:19 PM	1/2/2024	AUDIT	AUDIT A-129	1/1/2025

DATA RETENTION & DESTRUCTION (CONFIGURATION)

FIG. 25

2600

DATA RETENTION AND DESTRUCTION REPOSITORY: 1531-12575933

EDIT VIEW SAVE SAVE & CLOSE

MODULE INFORMATION

MODULE ID	MODULE NAME	DATE FIELD TO BE USED FOR RETENTION PERIOD	RETENTION PERIOD	RETENTION LEAD
1,531	CLIENT REDRESS	LAST UPDATED	BCH1000 - BRANCH TRANSACTION DATA AND SUMMARY REPORTS (1 YEAR)	DOE, JOHN

DATA RETENTION & DESTRUCTION

DRD ID: 1531-12575933 STATUS: ON HOLD

RECORD INFORMATION

MODULE NAME:	CLIENT REDRESS	LINK TO RECORD:	12575933
FIELD 1:	IKASDOFIJ	DATE FIELD:	
FIELD 2:	DRAFT	FIRST PUBLISHED DATE:	1/31/2023 4:39 AM
FIELD 3:	CR-BUPOC	LAST UPDATED DATE:	6/14/2023 11:03 AM

HOLD DETAILS

HOLD RECORD: ◉ YES ○ NO
EDIT

*HOLD REASON: ◉ AUDIT ○ LEGAL ○ OTHER ○ COMPLIANCE ○ MANAGER
EDIT

*DATE TO REVISIT: 7/21/2023

REFERENCE: AUDIT A-125

DATE PLACED ON HOLD: 7/11/2023

*HOLD JUSTIFICATION: 1234

SYSTEM INFORMATION

*REQUIRED

FIG. 26

2700

INVENTORY: DRD_TEST_APP1

EDIT VIEW | SAVE | SAVE & CLOSE

FIRST PUBLISHED: 7/17/2023 8:00 PM LAST UPDATED: 9/20/2024 6:32 PM

MODULE INFO | METADATA | DATA RETENTION & DESTRUCTION

DATA RETENTION & DESTRUCTION

RETENTION PERIOD:	BCH1000 - BRANCH LOGS (1 YEAR)	ENROLLMENT STATUS:	ENROLLED - ACTIVE
DATE FIELD TO BE USED FOR RETENTION PERIOD:	OTHER: RECENT INSURANCE RENEWAL DATE		
RETENTION LEAD:	SMITH, JOHN	APPLICATION FIELD TO DISPLAY 1:	NAME
		APPLICATION FIELD TO DISPLAY 2:	INSURANCE POLICY STATUS
		APPLICATION FIELD TO DISPLAY 3:	RECENT INSURANCE RENEWAL DATE

RECORDS MANAGEMENT | AUTHORIZATION OF DESTRUCTION | DESTROYED RECORDS

AUTHORIZATION OF DESTRUCTION

RECORDS TO BE AUTHORIZED FOR DESTRUCTION:

DISPLAY REPORT 1 TO 50 (OF 16,745)

DRD RECORD	LINK TO RECORD	FIELD 1	FIELD 2	FIELD 3	DATE FIELD	FIRST PUBLISHED DATE	LAST UPDATED DATE
1594-39837124	39837124	POLICY HOLDER1	ACTIVE	3/15/2022	3/14/2022 8:00 PM	7/17/2023 7:03 AM	7/17/2023 7:25 AM
1594-39837126	39837126	POLICY HOLDER1	ACTIVE	3/15/2022	3/14/2022 8:00 PM	7/17/2023 7:25 AM	7/17/2023 7:25 AM
1594-39837127	39837127	POLICY HOLDER1	INACTIVE	3/15/2022	3/14/2022 8:00 PM	7/17/2023 7:25 AM	7/17/2023 7:25 AM
1594-39837128	39837128	POLICY HOLDER1	INACTIVE	3/15/2022	3/14/2022 8:00 PM	7/17/2023 7:25 AM	7/17/2023 7:25 AM

RECORDS AUTHORIZED FOR DESTRUCTION

AUTHORIZE: ☐ I AUTHORIZE AUTOMATICALLY DELETING THE RECORDS LISTED IN THE BELOW REPORT

RECORDS AUTHORIZED FOR DESTRUCTION:

DISPLAY REPORT 0 TO 0 (OF 0)

DRD RECORD	LINK TO RECORD	FIELD 1	FIELD 2	FIELD 3	DATE FIELD	FIRST PUBLISHED DATE	LAST UPDATED DATE

NO RECORDS FOUND

PAGE 1 OF 1 (0 RECORDS)

FIG. 27

2800

INVENTORY: CLIENT REDRESS

EDIT VIEW | SAVE | SAVE & CLOSE

FIRST PUBLISHED: 11/16/2022 8:00 PM LAST UPDATED: 12/3/2024 8:33 AM

MODULE INFO | METADATA | DATA RETENTION & DESTRUCTION

▿ DATA RETENTION & DESTRUCTION

RETENTION PERIOD:	BCH1000 - BRANCH TRANSACTION DATA AND SUMMARY REPORTS (1 YEAR)	ENROLLMENT STATUS:	ENROLLED - ACTIVE
DATE FIELD TO BE USED FOR RETENTION PERIOD:	LAST UPDATED		
RETENTION LEAD:	DOE, JOHN	APPLICATION FIELD TO DISPLAY 1:	TITLE
		APPLICATION FIELD TO DISPLAY 2:	STATUS
		APPLICATION FIELD TO DISPLAY 3:	BU POC

RECORDS MANAGEMENT | AUTHORIZATION OF DESTRUCTION | DESTROYED RECORDS

▿ DESTROYED RECORDS

DESTROYED RECORDS:

DISPLAY REPORT 1 TO 18 (OF 18)

DRD RECORD	LINK TO RECORD	AUTHORIZATION DATE	DESTRUCTION DATE/TIME	
1531-12531509	12531509	7/11/2023	7/11/2023 3:29 PM	
1531-12534298	12534298	7/11/2023	7/11/2023 3:30 PM	
1531-12534417	12534417	7/11/2023	7/11/2023 3:30 PM	
1531-12534421	12534421	7/11/2023	7/11/2023 3:30 PM	
1531-12534435	12534435	7/11/2023	7/11/2023 3:30 PM	
1531-12534568	12534568	7/11/2023	7/11/2023 3:30 PM	
1531-12535033	12535033	7/11/2023	7/11/2023 3:30 PM	

▿ DATA RETENTION & DESTRUCTION (CONFIGURATION)

◦ REQUIRED

FIG. 28

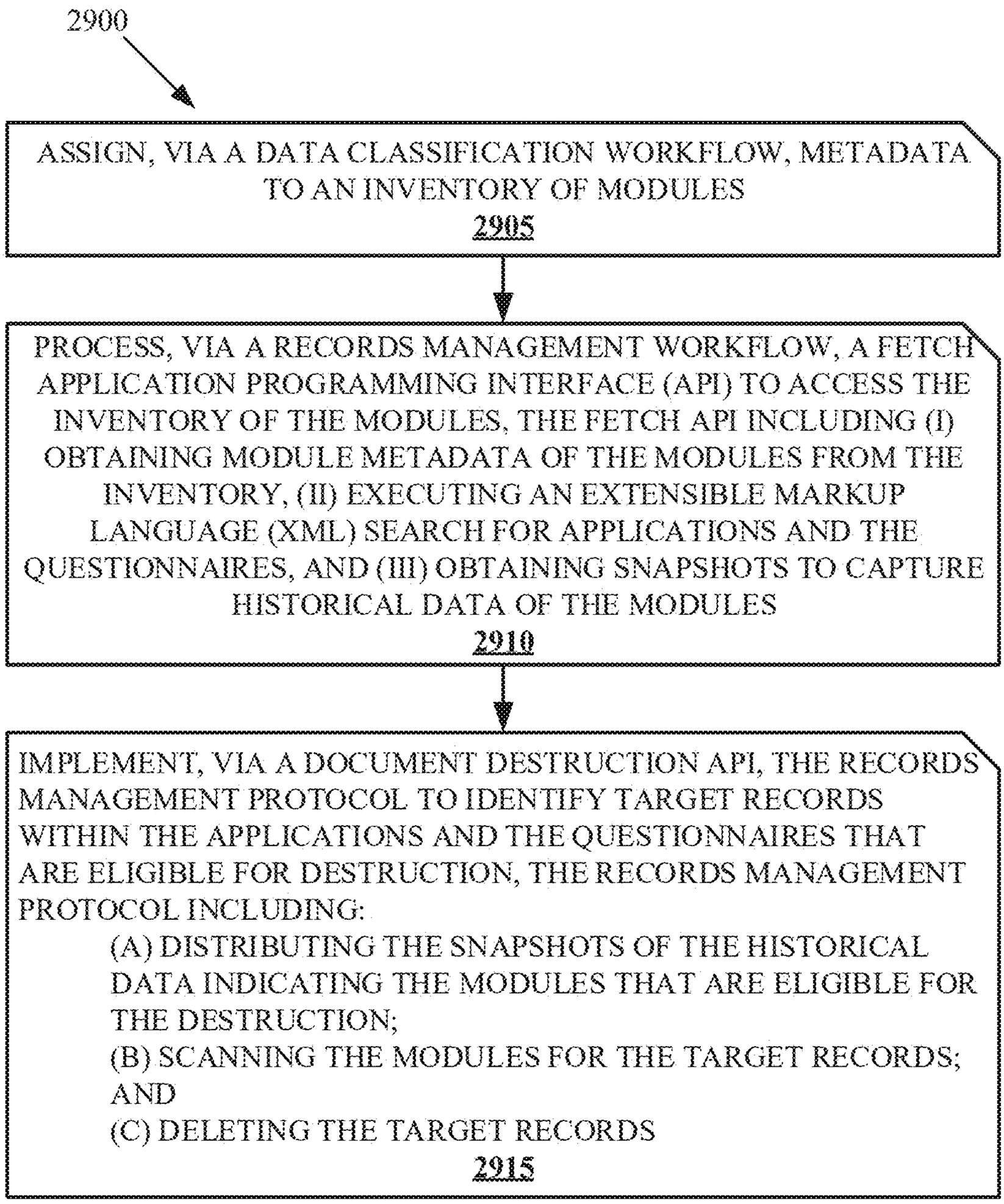

2900
ASSIGN, VIA A DATA CLASSIFICATION WORKFLOW, METADATA TO AN INVENTORY OF MODULES
2905
PROCESS, VIA A RECORDS MANAGEMENT WORKFLOW, A FETCH APPLICATION PROGRAMMING INTERFACE (API) TO ACCESS THE INVENTORY OF THE MODULES, THE FETCH API INCLUDING (I) OBTAINING MODULE METADATA OF THE MODULES FROM THE INVENTORY, (II) EXECUTING AN EXTENSIBLE MARKUP LANGUAGE (XML) SEARCH FOR APPLICATIONS AND THE QUESTIONNAIRES, AND (III) OBTAINING SNAPSHOTS TO CAPTURE HISTORICAL DATA OF THE MODULES
2910
IMPLEMENT, VIA A DOCUMENT DESTRUCTION API, THE RECORDS MANAGEMENT PROTOCOL TO IDENTIFY TARGET RECORDS WITHIN THE APPLICATIONS AND THE QUESTIONNAIRES THAT ARE ELIGIBLE FOR DESTRUCTION, THE RECORDS MANAGEMENT PROTOCOL INCLUDING:
(A) DISTRIBUTING THE SNAPSHOTS OF THE HISTORICAL DATA INDICATING THE MODULES THAT ARE ELIGIBLE FOR THE DESTRUCTION;
(B) SCANNING THE MODULES FOR THE TARGET RECORDS; AND
(C) DELETING THE TARGET RECORDS
2915

FIG. 29

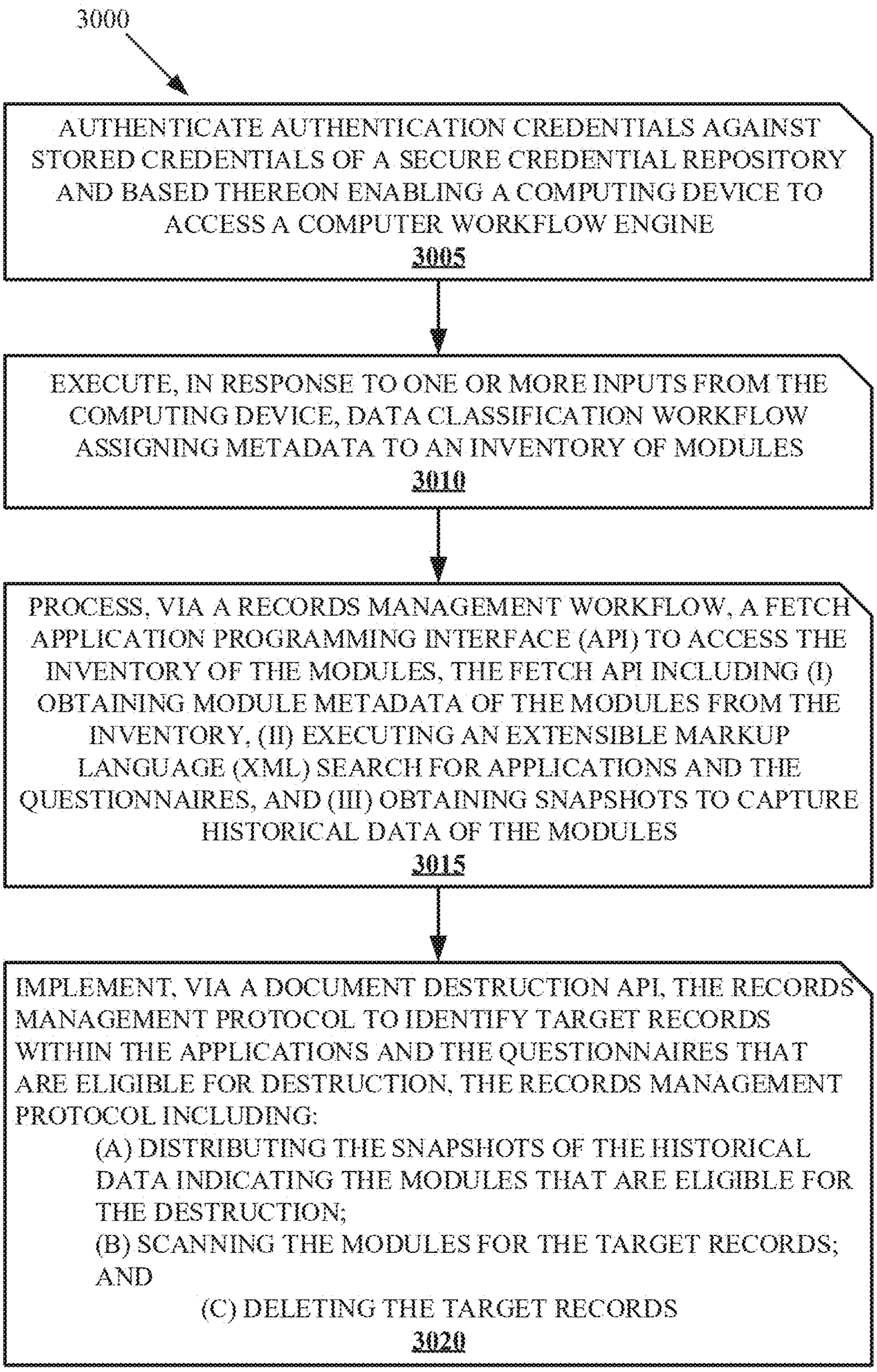

3000
AUTHENTICATE AUTHENTICATION CREDENTIALS AGAINST STORED CREDENTIALS OF A SECURE CREDENTIAL REPOSITORY AND BASED THEREON ENABLING A COMPUTING DEVICE TO ACCESS A COMPUTER WORKFLOW ENGINE
3005
EXECUTE, IN RESPONSE TO ONE OR MORE INPUTS FROM THE COMPUTING DEVICE, DATA CLASSIFICATION WORKFLOW ASSIGNING METADATA TO AN INVENTORY OF MODULES
3010
PROCESS, VIA A RECORDS MANAGEMENT WORKFLOW, A FETCH APPLICATION PROGRAMMING INTERFACE (API) TO ACCESS THE INVENTORY OF THE MODULES, THE FETCH API INCLUDING (I) OBTAINING MODULE METADATA OF THE MODULES FROM THE INVENTORY, (II) EXECUTING AN EXTENSIBLE MARKUP LANGUAGE (XML) SEARCH FOR APPLICATIONS AND THE QUESTIONNAIRES, AND (III) OBTAINING SNAPSHOTS TO CAPTURE HISTORICAL DATA OF THE MODULES
3015
IMPLEMENT, VIA A DOCUMENT DESTRUCTION API, THE RECORDS MANAGEMENT PROTOCOL TO IDENTIFY TARGET RECORDS WITHIN THE APPLICATIONS AND THE QUESTIONNAIRES THAT ARE ELIGIBLE FOR DESTRUCTION, THE RECORDS MANAGEMENT PROTOCOL INCLUDING:
(A) DISTRIBUTING THE SNAPSHOTS OF THE HISTORICAL DATA INDICATING THE MODULES THAT ARE ELIGIBLE FOR THE DESTRUCTION;
(B) SCANNING THE MODULES FOR THE TARGET RECORDS; AND
(C) DELETING THE TARGET RECORDS
3020

FIG. 30

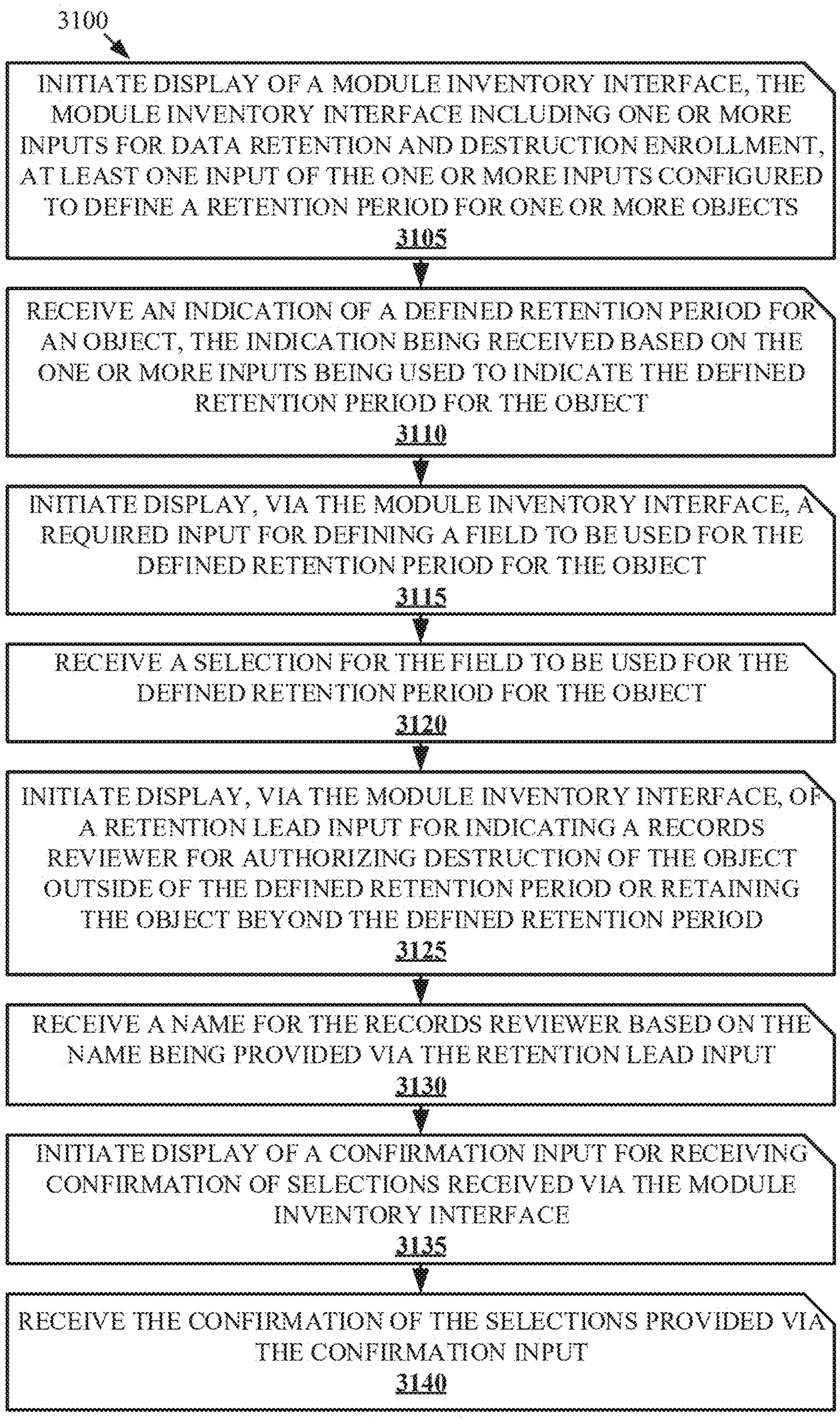

3100
INITIATE DISPLAY OF A MODULE INVENTORY INTERFACE, THE MODULE INVENTORY INTERFACE INCLUDING ONE OR MORE INPUTS FOR DATA RETENTION AND DESTRUCTION ENROLLMENT, AT LEAST ONE INPUT OF THE ONE OR MORE INPUTS CONFIGURED TO DEFINE A RETENTION PERIOD FOR ONE OR MORE OBJECTS
3105
RECEIVE AN INDICATION OF A DEFINED RETENTION PERIOD FOR AN OBJECT, THE INDICATION BEING RECEIVED BASED ON THE ONE OR MORE INPUTS BEING USED TO INDICATE THE DEFINED RETENTION PERIOD FOR THE OBJECT
3110
INITIATE DISPLAY, VIA THE MODULE INVENTORY INTERFACE, A REQUIRED INPUT FOR DEFINING A FIELD TO BE USED FOR THE DEFINED RETENTION PERIOD FOR THE OBJECT
3115
RECEIVE A SELECTION FOR THE FIELD TO BE USED FOR THE DEFINED RETENTION PERIOD FOR THE OBJECT
3120
INITIATE DISPLAY, VIA THE MODULE INVENTORY INTERFACE, OF A RETENTION LEAD INPUT FOR INDICATING A RECORDS REVIEWER FOR AUTHORIZING DESTRUCTION OF THE OBJECT OUTSIDE OF THE DEFINED RETENTION PERIOD OR RETAINING THE OBJECT BEYOND THE DEFINED RETENTION PERIOD
3125
RECEIVE A NAME FOR THE RECORDS REVIEWER BASED ON THE NAME BEING PROVIDED VIA THE RETENTION LEAD INPUT
3130
INITIATE DISPLAY OF A CONFIRMATION INPUT FOR RECEIVING CONFIRMATION OF SELECTIONS RECEIVED VIA THE MODULE INVENTORY INTERFACE
3135
RECEIVE THE CONFIRMATION OF THE SELECTIONS PROVIDED VIA THE CONFIRMATION INPUT
3140

FIG. 31

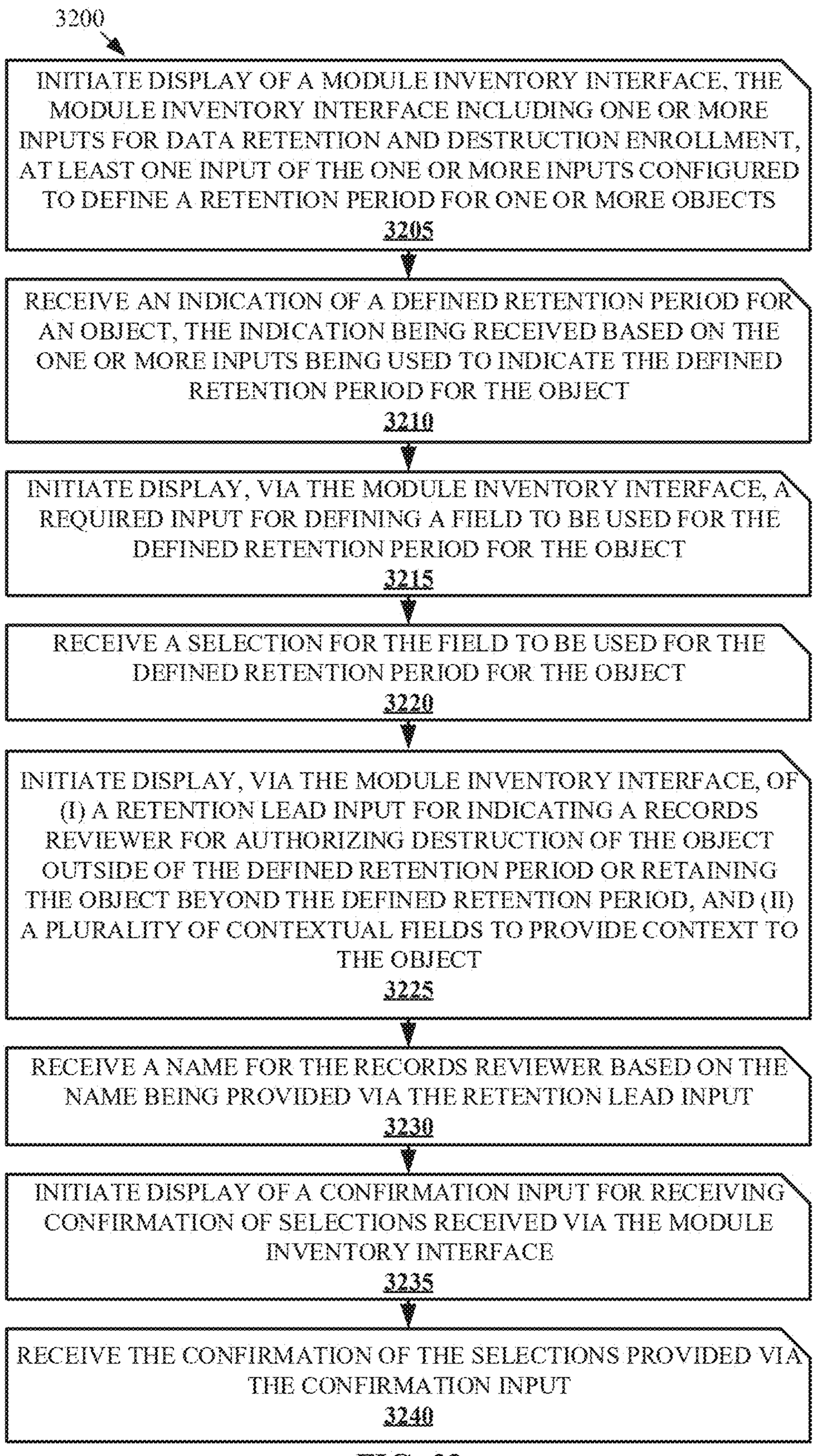

3200
INITIATE DISPLAY OF A MODULE INVENTORY INTERFACE, THE MODULE INVENTORY INTERFACE INCLUDING ONE OR MORE INPUTS FOR DATA RETENTION AND DESTRUCTION ENROLLMENT, AT LEAST ONE INPUT OF THE ONE OR MORE INPUTS CONFIGURED TO DEFINE A RETENTION PERIOD FOR ONE OR MORE OBJECTS
3205
RECEIVE AN INDICATION OF A DEFINED RETENTION PERIOD FOR AN OBJECT, THE INDICATION BEING RECEIVED BASED ON THE ONE OR MORE INPUTS BEING USED TO INDICATE THE DEFINED RETENTION PERIOD FOR THE OBJECT
3210
INITIATE DISPLAY, VIA THE MODULE INVENTORY INTERFACE, A REQUIRED INPUT FOR DEFINING A FIELD TO BE USED FOR THE DEFINED RETENTION PERIOD FOR THE OBJECT
3215
RECEIVE A SELECTION FOR THE FIELD TO BE USED FOR THE DEFINED RETENTION PERIOD FOR THE OBJECT
3220
INITIATE DISPLAY, VIA THE MODULE INVENTORY INTERFACE, OF (I) A RETENTION LEAD INPUT FOR INDICATING A RECORDS REVIEWER FOR AUTHORIZING DESTRUCTION OF THE OBJECT OUTSIDE OF THE DEFINED RETENTION PERIOD OR RETAINING THE OBJECT BEYOND THE DEFINED RETENTION PERIOD, AND (II) A PLURALITY OF CONTEXTUAL FIELDS TO PROVIDE CONTEXT TO THE OBJECT
3225
RECEIVE A NAME FOR THE RECORDS REVIEWER BASED ON THE NAME BEING PROVIDED VIA THE RETENTION LEAD INPUT
3230
INITIATE DISPLAY OF A CONFIRMATION INPUT FOR RECEIVING CONFIRMATION OF SELECTIONS RECEIVED VIA THE MODULE INVENTORY INTERFACE
3235
RECEIVE THE CONFIRMATION OF THE SELECTIONS PROVIDED VIA THE CONFIRMATION INPUT
3240

FIG. 32

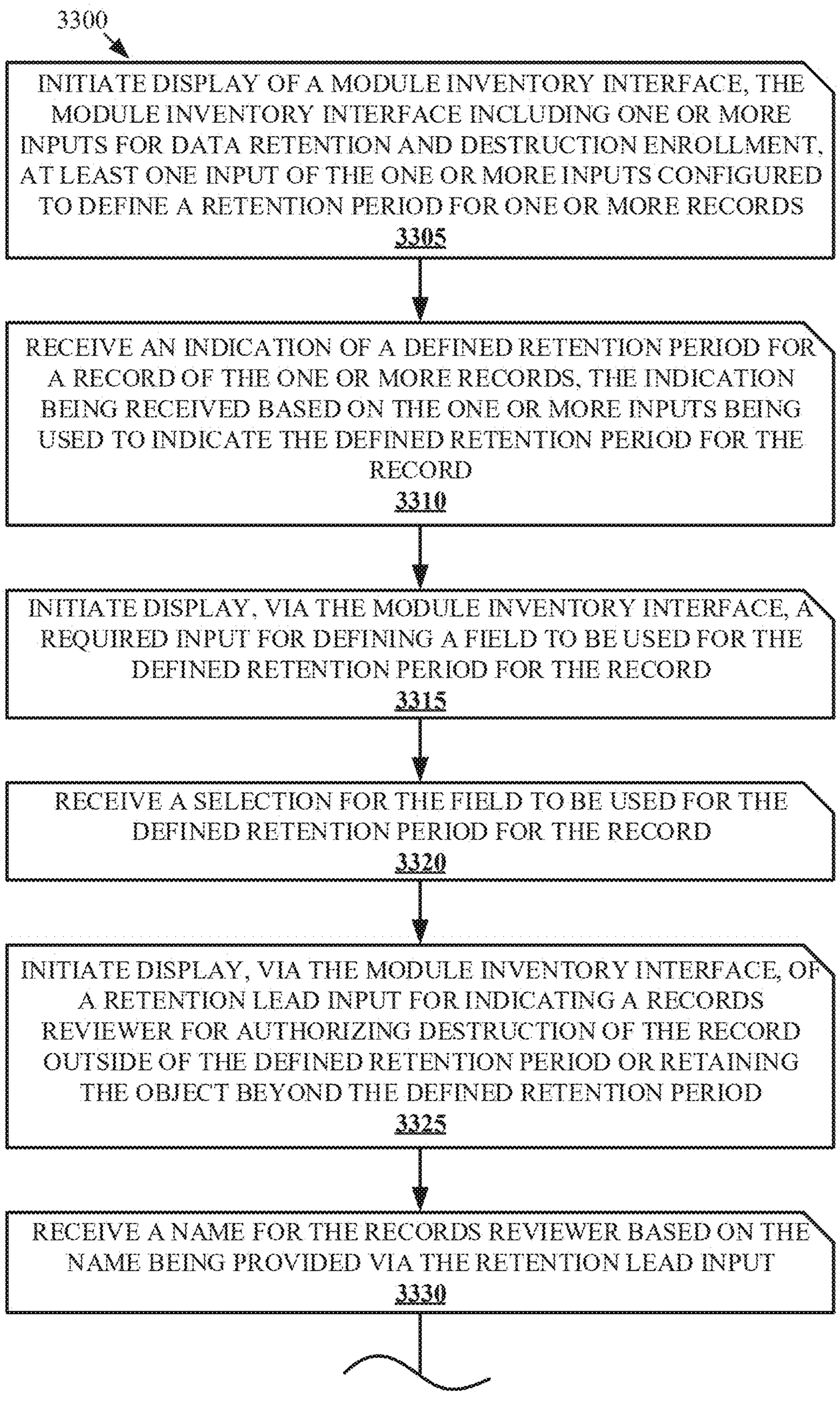

3300
INITIATE DISPLAY OF A MODULE INVENTORY INTERFACE, THE MODULE INVENTORY INTERFACE INCLUDING ONE OR MORE INPUTS FOR DATA RETENTION AND DESTRUCTION ENROLLMENT, AT LEAST ONE INPUT OF THE ONE OR MORE INPUTS CONFIGURED TO DEFINE A RETENTION PERIOD FOR ONE OR MORE RECORDS
3305
RECEIVE AN INDICATION OF A DEFINED RETENTION PERIOD FOR A RECORD OF THE ONE OR MORE RECORDS, THE INDICATION BEING RECEIVED BASED ON THE ONE OR MORE INPUTS BEING USED TO INDICATE THE DEFINED RETENTION PERIOD FOR THE RECORD
3310
INITIATE DISPLAY, VIA THE MODULE INVENTORY INTERFACE, A REQUIRED INPUT FOR DEFINING A FIELD TO BE USED FOR THE DEFINED RETENTION PERIOD FOR THE RECORD
3315
RECEIVE A SELECTION FOR THE FIELD TO BE USED FOR THE DEFINED RETENTION PERIOD FOR THE RECORD
3320
INITIATE DISPLAY, VIA THE MODULE INVENTORY INTERFACE, OF A RETENTION LEAD INPUT FOR INDICATING A RECORDS REVIEWER FOR AUTHORIZING DESTRUCTION OF THE RECORD OUTSIDE OF THE DEFINED RETENTION PERIOD OR RETAINING THE OBJECT BEYOND THE DEFINED RETENTION PERIOD
3325
RECEIVE A NAME FOR THE RECORDS REVIEWER BASED ON THE NAME BEING PROVIDED VIA THE RETENTION LEAD INPUT
3330

FIG. 33A

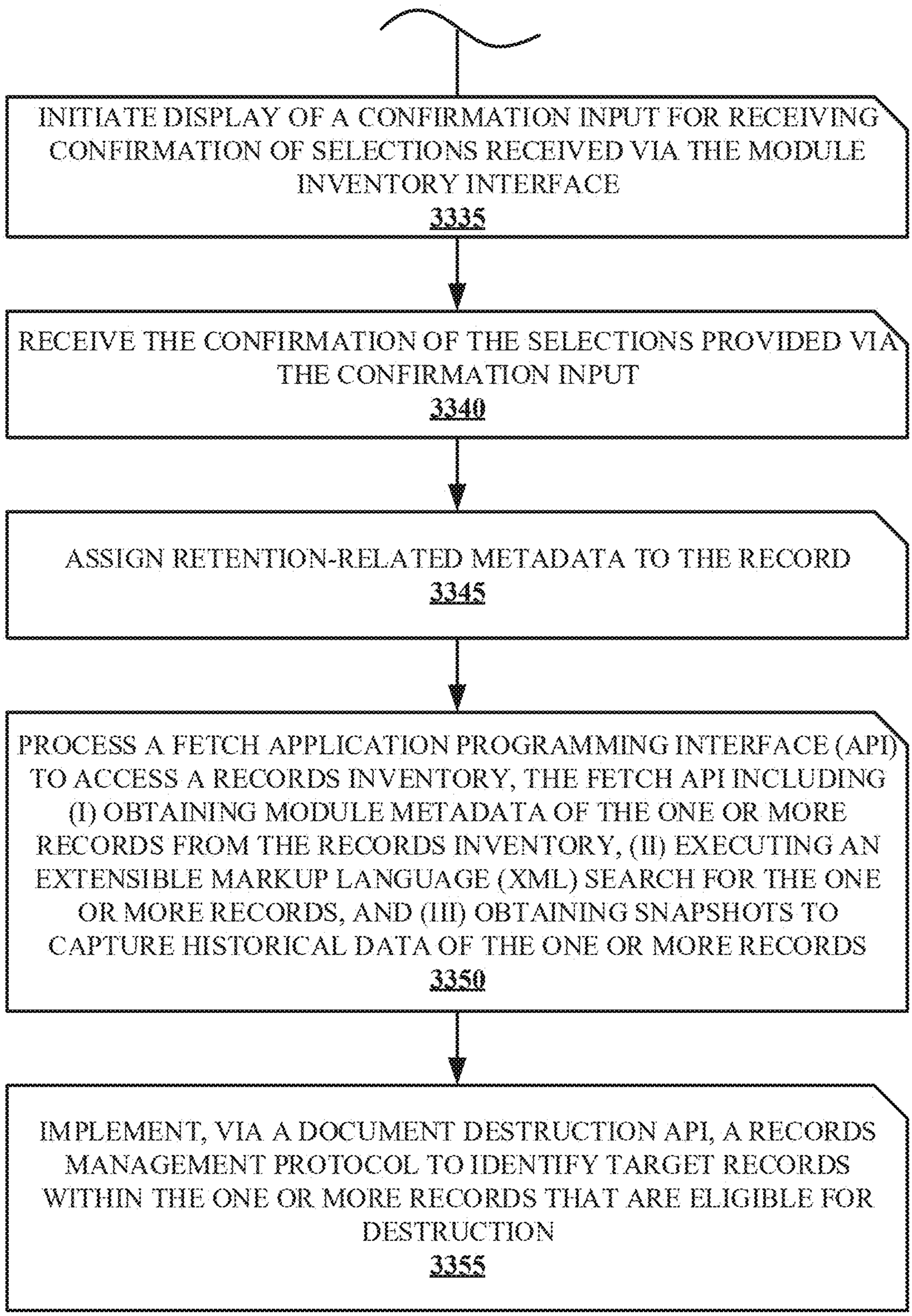

INITIATE DISPLAY OF A CONFIRMATION INPUT FOR RECEIVING CONFIRMATION OF SELECTIONS RECEIVED VIA THE MODULE INVENTORY INTERFACE
3335
RECEIVE THE CONFIRMATION OF THE SELECTIONS PROVIDED VIA THE CONFIRMATION INPUT
3340
ASSIGN RETENTION-RELATED METADATA TO THE RECORD
3345
PROCESS A FETCH APPLICATION PROGRAMMING INTERFACE (API) TO ACCESS A RECORDS INVENTORY, THE FETCH API INCLUDING (I) OBTAINING MODULE METADATA OF THE ONE OR MORE RECORDS FROM THE RECORDS INVENTORY, (II) EXECUTING AN EXTENSIBLE MARKUP LANGUAGE (XML) SEARCH FOR THE ONE OR MORE RECORDS, AND (III) OBTAINING SNAPSHOTS TO CAPTURE HISTORICAL DATA OF THE ONE OR MORE RECORDS
3350
IMPLEMENT, VIA A DOCUMENT DESTRUCTION API, A RECORDS MANAGEMENT PROTOCOL TO IDENTIFY TARGET RECORDS WITHIN THE ONE OR MORE RECORDS THAT ARE ELIGIBLE FOR DESTRUCTION
3355

FIG. 33B

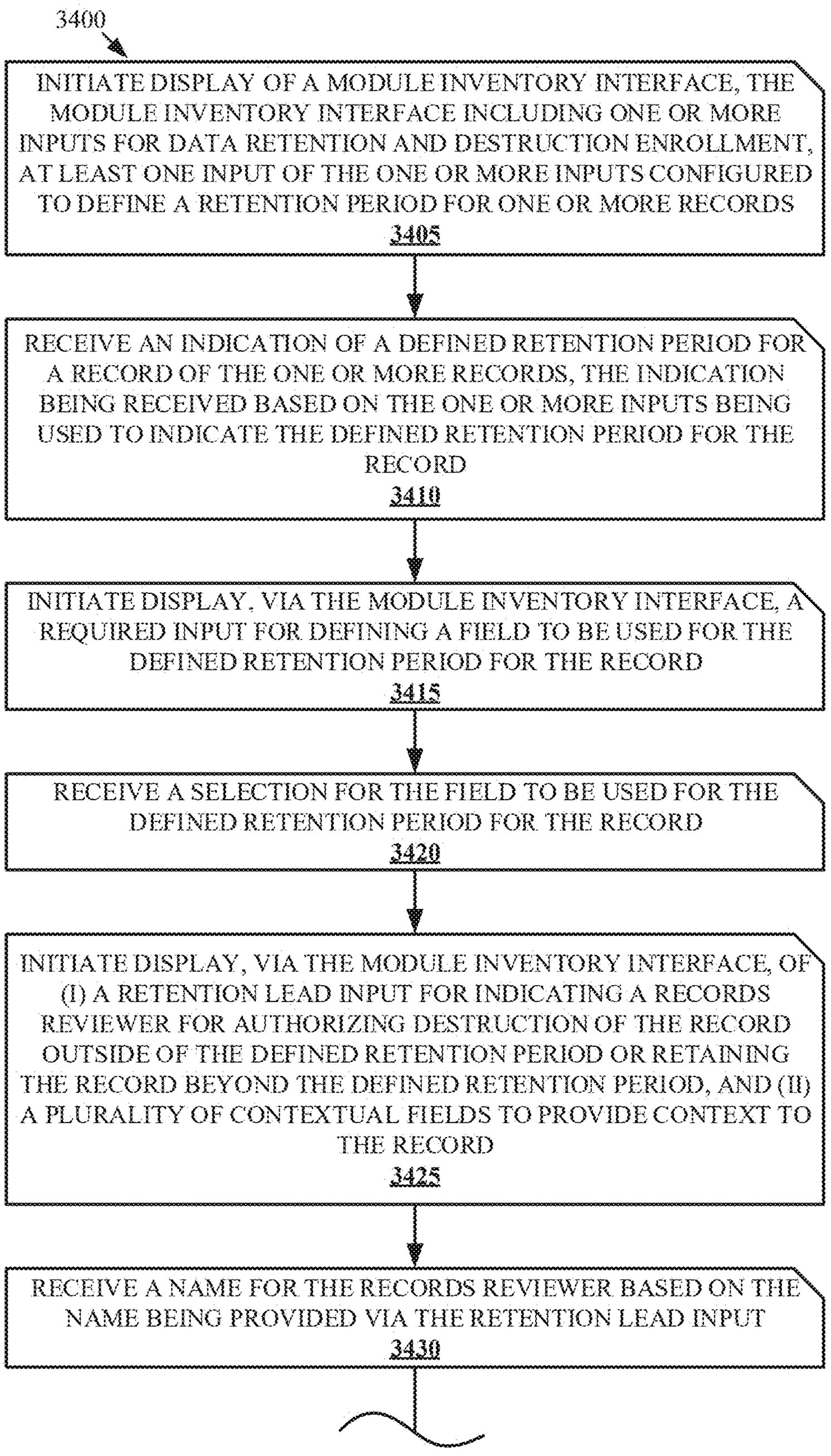

3400
INITIATE DISPLAY OF A MODULE INVENTORY INTERFACE, THE MODULE INVENTORY INTERFACE INCLUDING ONE OR MORE INPUTS FOR DATA RETENTION AND DESTRUCTION ENROLLMENT, AT LEAST ONE INPUT OF THE ONE OR MORE INPUTS CONFIGURED TO DEFINE A RETENTION PERIOD FOR ONE OR MORE RECORDS
3405
RECEIVE AN INDICATION OF A DEFINED RETENTION PERIOD FOR A RECORD OF THE ONE OR MORE RECORDS, THE INDICATION BEING RECEIVED BASED ON THE ONE OR MORE INPUTS BEING USED TO INDICATE THE DEFINED RETENTION PERIOD FOR THE RECORD
3410
INITIATE DISPLAY, VIA THE MODULE INVENTORY INTERFACE, A REQUIRED INPUT FOR DEFINING A FIELD TO BE USED FOR THE DEFINED RETENTION PERIOD FOR THE RECORD
3415
RECEIVE A SELECTION FOR THE FIELD TO BE USED FOR THE DEFINED RETENTION PERIOD FOR THE RECORD
3420
INITIATE DISPLAY, VIA THE MODULE INVENTORY INTERFACE, OF (I) A RETENTION LEAD INPUT FOR INDICATING A RECORDS REVIEWER FOR AUTHORIZING DESTRUCTION OF THE RECORD OUTSIDE OF THE DEFINED RETENTION PERIOD OR RETAINING THE RECORD BEYOND THE DEFINED RETENTION PERIOD, AND (II) A PLURALITY OF CONTEXTUAL FIELDS TO PROVIDE CONTEXT TO THE RECORD
3425
RECEIVE A NAME FOR THE RECORDS REVIEWER BASED ON THE NAME BEING PROVIDED VIA THE RETENTION LEAD INPUT
3430

FIG. 34A

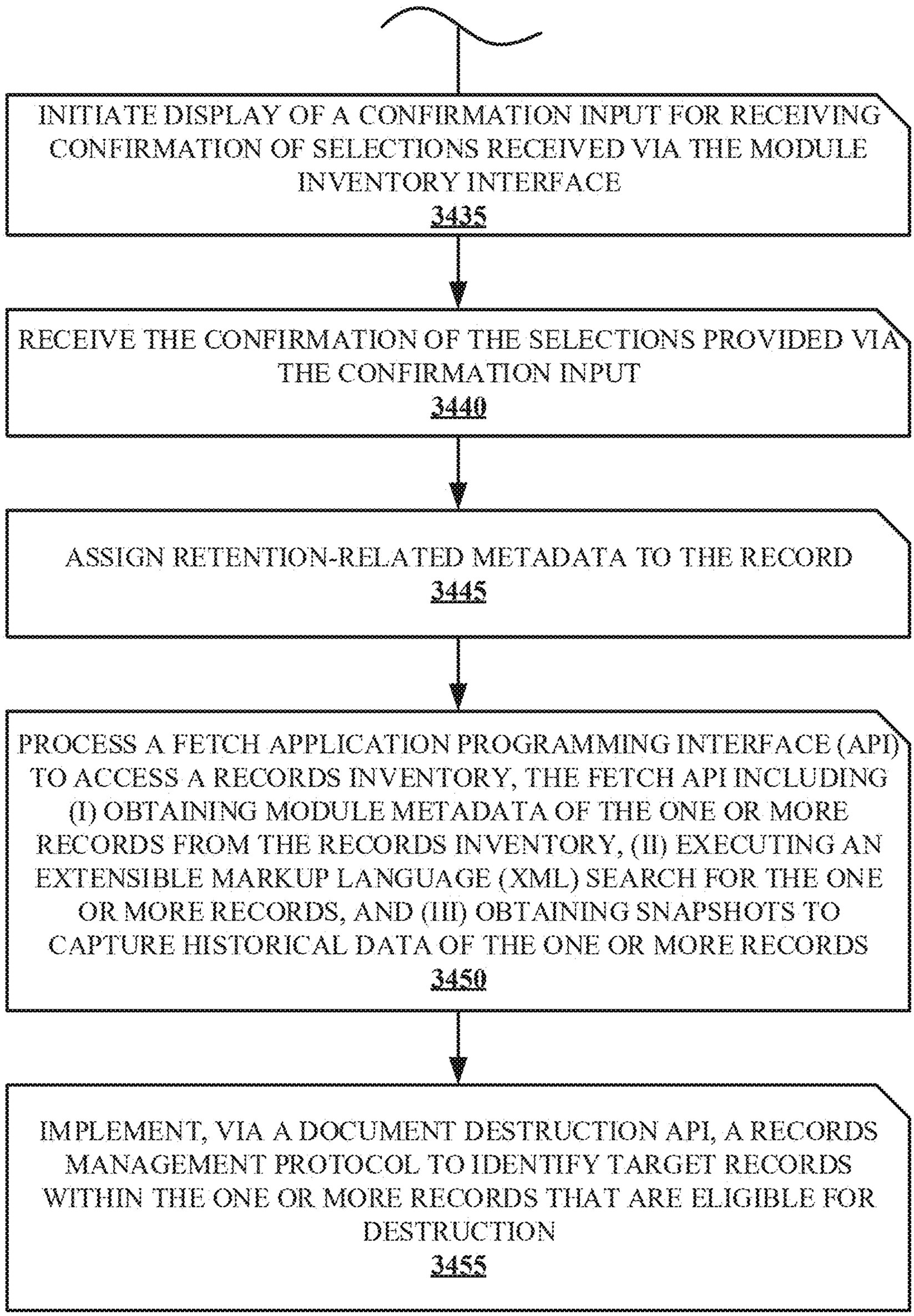

INITIATE DISPLAY OF A CONFIRMATION INPUT FOR RECEIVING CONFIRMATION OF SELECTIONS RECEIVED VIA THE MODULE INVENTORY INTERFACE
3435
RECEIVE THE CONFIRMATION OF THE SELECTIONS PROVIDED VIA THE CONFIRMATION INPUT
3440
ASSIGN RETENTION-RELATED METADATA TO THE RECORD
3445
PROCESS A FETCH APPLICATION PROGRAMMING INTERFACE (API) TO ACCESS A RECORDS INVENTORY, THE FETCH API INCLUDING (I) OBTAINING MODULE METADATA OF THE ONE OR MORE RECORDS FROM THE RECORDS INVENTORY, (II) EXECUTING AN EXTENSIBLE MARKUP LANGUAGE (XML) SEARCH FOR THE ONE OR MORE RECORDS, AND (III) OBTAINING SNAPSHOTS TO CAPTURE HISTORICAL DATA OF THE ONE OR MORE RECORDS
3450
IMPLEMENT, VIA A DOCUMENT DESTRUCTION API, A RECORDS MANAGEMENT PROTOCOL TO IDENTIFY TARGET RECORDS WITHIN THE ONE OR MORE RECORDS THAT ARE ELIGIBLE FOR DESTRUCTION
3455

FIG. 34B

DATABASE RETENTION AND DESTRUCTION SYSTEMS FOR RECORDS MANAGEMENT

FIELD OF THE INVENTION

This invention relates generally to the field of records management systems, and more particularly embodiments of the invention relate to data retention and destruction systems for databases.

BACKGROUND OF THE INVENTION

Deficiencies in data retention processes can lead to significant risk for an organization. Specifically, such deficiencies can lead to an organization being noncompliant with governmental regulations and can expose the databases to security vulnerabilities. In particular, financial institutions must comply with strict governmental regulations related to data retention of financial records. In addition, as data accumulates this can cause lag in computing operations and can take up unnecessary storage space. Existing processes used by organizations for data retention and destruction can be inconsistent, can become outdated, and often result in irrelevant and excessive data retention, which can increase storage costs and make it more difficult to locate important information. A lack of adherence to organizational policies and governmental regulations can also lead some employees to inadvertently retaining sensitive and confidential information beyond the intended lifespan, which can expose the organization to legal issues. For example, retaining data longer than is permitted under governmental regulations can result in significant fines and legal exposure for the organization. Retention of unnecessary data introduces additional problems in the event of a data breach, as some information may be expose that should not have been susceptible to breach. A failure to incorporate automated data retention and destruction systems for records management can introduce manual errors and inconsistent application of governmental regulations and organizational policies. Thus, a need exists for improved systems and methods for data retention and destruction management.

BRIEF SUMMARY

Shortcomings of the prior art are overcome and additional advantages are provided through the provision of a computing system for implementing a records management protocol controlling data retention and destruction. The system includes, for instance, at least one processor, a communication interface communicatively coupled to the at least one processor, and a memory device storing executable code that is executable by the at least one processor. Execution of the executable code causes the at least one processor to initiate display of a module inventory interface, the module inventory interface including one or more inputs for data retention and destruction enrollment, at least one input of the one or more inputs configured to define a retention period for one or more objects. The system also receives an indication of a defined retention period for an object, the indication being received based on the one or more inputs being used to indicate the defined retention period for the object. Further, the system initiates display, via the module inventory interface, a required input for defining a field to be used for the defined retention period for the object. The system receives receive a selection for the field to be used for the defined retention period for the object and initiates display, via the module inventory interface, of a retention lead input for indicating a records reviewer for authorizing destruction of the object outside of the defined retention period or retaining the object beyond the defined retention period. The system receives a name for the records reviewer based on the name being provided via the retention lead input and initiates display of a confirmation input for receiving confirmation of selections received via the module inventory interface. In addition, the system receives the confirmation of the selections provided via the confirmation input.

Additionally, disclosed herein is a computing system that includes at least one processor, a communication interface communicatively coupled to the at least one processor, and a memory device storing executable code. When executed, the executable code causes the at least one processor to initiate display of a module inventory interface, the module inventory interface including one or more inputs for data retention and destruction enrollment, at least one input of the one or more inputs configured to define a retention period for one or more objects. The system receives an indication of a defined retention period for an object, the indication being received based on the one or more inputs being used to indicate the defined retention period for the object. Further, the system initiates display, via the module inventory interface, a required input for defining a field to be used for the defined retention period for the object. The system also receives a selection for the field to be used for the defined retention period for the object and initiates display, via the module inventory interface, of (i) a retention lead input for indicating a records reviewer for authorizing destruction of the object outside of the defined retention period or retaining the object beyond the defined retention period, and (ii) a plurality of contextual fields to provide context to the object. In addition, the system receives a name for the records reviewer based on the name being provided via the retention lead input, and the system initiates display of a confirmation input for receiving confirmation of selections received via the module inventory interface. Further, the confirmation of the selections provided via the confirmation input is received.

Also disclosed herein is a computer-implemented method that includes initiating display of a module inventory interface, the module inventory interface including one or more inputs for data retention and destruction enrollment, at least one input of the one or more inputs configured to define a retention period for one or more objects. The method also includes receiving an indication of a defined retention period for an object, the indication being received based on the one or more inputs being used to indicate the defined retention period for the object. Further, the method includes initiating display, via the module inventory interface, a required input for defining a field to be used for the defined retention period for the object and receiving a selection for the field to be used for the defined retention period for the object. In addition, the method includes initiating display, via the module inventory interface, of a retention lead input for indicating a records reviewer for authorizing destruction of the object outside of the defined retention period or retaining the object beyond the defined retention period. The method also receives a name for the records reviewer based on the name being provided via the retention lead input, and initiates display of a confirmation input for receiving confirmation of selections received via the module inventory interface. The confirmation of the selections provided via the confirmation input is then received.

The features, functions, and advantages that have been described herein may be achieved independently in various

embodiments of the present invention including computer-implemented methods, computer program products, and computing systems or may be combined in yet other embodiments, further details of which can be seen with reference to the following description and drawings.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

One or more aspects are particularly pointed out and distinctly claimed as examples in the claims at the conclusion of the specification. The foregoing as well as objects, features, and advantages of one or more aspects are apparent from the following detailed description taken in conjunction with the accompanying drawings in which:

FIG. **1** illustrates an enterprise system, and environment thereof for software control systems, in accordance with an embodiment of the present invention;

FIG. **2**A is a diagram of a feedforward network, according to at least one embodiment, utilized in machine learning;

FIG. **2**B is a diagram of a convolution neural network, according to at least one embodiment, utilized in machine learning;

FIG. **2**C is a diagram of a portion of the convolution neural network of FIG. **2**B, according to at least one embodiment, illustrating assigned weights at connections or neurons;

FIG. **3** is a diagram representing an exemplary weighted sum computation in a node in an artificial neural network;

FIG. **4** is a diagram of a Recurrent Neural Network RNN, according to at least one embodiment, utilized in machine learning;

FIG. **5** is a schematic logic diagram of an artificial intelligence program including a front-end and a back-end algorithm;

FIG. **6** is a flow chart representing a method, according to at least one embodiment, of model development and deployment by machine learning;

FIG. **7** depicts an example architecture for data retention and destruction, according to an embodiment of the present invention;

FIG. **8** depicts module inventory information and a front-end record setup interface associated with the module inventory, in accordance with an embodiment of the present invention;

FIG. **9** depicts a front-end interface for a date field used for data retention that is associated with the module inventory, in accordance with an embodiment of the present invention;

FIG. **10** depicts additional information included in the front-end interface of FIG. **9** for specifying a retention lead, in accordance with an embodiment of the present invention;

FIG. **11** depicts a front-end interface for confirming enrollment associated with a data retention and destruction settings, in accordance with an embodiment of the present invention;

FIG. **12** depicts a front-end interface for administrative setup associated with the module inventory, in accordance with an embodiment of the present invention;

FIG. **13** depicts a front-end interface associated with retention lead and fields to display that is associated with the module inventory, in accordance with an embodiment of the present invention;

FIG. **14** depicts data classification inventory information, in accordance with an embodiment of the present invention;

FIG. **15** depicts a front-end interface for selecting data classification associated with the data classification inventory, in accordance with an embodiment of the present invention;

FIG. **16** depicts records management information associated with records management, in accordance with an embodiment of the present invention;

FIG. **17** depicts a front-end interface depicting records pending decision that is associated with records management, in accordance with an embodiment of the present invention;

FIG. **18** depicts a front-end interface depicting information necessary for placing a record on hold, in accordance with an embodiment of the present invention;

FIG. **19** depicts a front-end interface depicting records that are on hold, in accordance with an embodiment of the present invention;

FIG. **20** depicts a front-end interface for authorizing destruction of records, in accordance with an embodiment of the present invention;

FIG. **21** depicts a front-end interface for logging destroyed records, in accordance with an embodiment of the present invention;

FIG. **22** depicts an example data classification workflow facilitating records management, in accordance with an embodiment of the present invention;

FIG. **23** depicts a front-end interface associated with tag snapshots of records that have been locked and authorized for destruction, in accordance with an embodiment of the present invention;

FIG. **24** depicts a front-end interface for data retention and destruction reporting, in accordance with an embodiment of the present invention;

FIG. **25** depicts a front-end interface for accessing reports associated with data retention and destruction, in accordance with the present invention;

FIG. **26** depicts a front-end interface associated with a data retention and destruction repository, in accordance with the present invention;

FIG. **27** depicts an example report associated with data retention and destruction, in accordance with the present invention;

FIG. **28** depicts an example report of destroyed records, in accordance with the present invention;

FIG. **29** depicts a block diagram of an example method, in accordance with an embodiment of the present invention;

FIG. **30** depicts a block diagram of an example method, in accordance with an embodiment of the present invention;

FIG. **31** depicts a block diagram of an example method, in accordance with an embodiment of the present invention;

FIG. **32** depicts a block diagram of an example method, in accordance with an embodiment of the present invention;

FIG. **33**A depicts a block diagram of an example method, in accordance with an embodiment of the present invention;

FIG. **33**B depicts additional aspects of the example method of FIG. **33**A,

FIG. **34**A depicts a block diagram of an example method, in accordance with an embodiment of the present invention; and

FIG. **34**B depicts additional aspects of the example method of FIG. **34**A.

DETAILED DESCRIPTION

Aspects of the present invention and certain features, advantages, and details thereof are explained more fully below with reference to the non-limiting examples illus-

trated in the accompanying drawings. Descriptions of well-known processing techniques, systems, components, etc. are omitted so as to not unnecessarily obscure the invention in detail. It should be understood that the detailed description and the specific examples, while indicating aspects of the invention, are given by way of illustration only, and not by way of limitation. Various substitutions, modifications, additions, and/or arrangements, within the spirit and/or scope of the underlying inventive concepts will be apparent to those skilled in the art from this disclosure. Note further that numerous inventive aspects and features are disclosed herein, and unless inconsistent, each disclosed aspect or feature is combinable with any other disclosed aspect or feature as desired for a particular embodiment of the concepts disclosed herein.

Unless described or implied as exclusive alternatives, features throughout the drawings and descriptions should be taken as cumulative, such that features expressly associated with some particular embodiments can be combined with other embodiments.

While certain exemplary embodiments have been described and shown in the accompanying drawings, it is to be understood that such embodiments are merely illustrative of, and not restrictive on, the broad invention, and that this invention not be limited to the specific constructions and arrangements shown and described, since various other changes, combinations, omissions, modifications and substitutions, in addition to those set forth in the above paragraphs, are possible. Those skilled in the art will appreciate that various adaptations, modifications, and combinations of the herein described embodiments can be configured without departing from the scope and spirit of the invention. Therefore, it is to be understood that, within the scope of the included claims, the invention may be practiced other than as specifically described herein.

Additionally, illustrative embodiments are described below using specific code, designs, architectures, protocols, layouts, schematics, or tools only as examples, and not by way of limitation. Furthermore, the illustrative embodiments are described in certain instances using particular software, tools, or data processing environments only as example for clarity of description. The illustrative embodiments can be used in conjunction with other comparable or similarly purposed structures, systems, applications, or architectures. One or more aspects of an illustrative embodiment can be implemented in hardware, software, or a combination thereof.

As understood by one skilled in the art, program code, as referred to in this application, can include both software and hardware. For example, program code in certain embodiments of the present invention can include fixed function hardware, while other embodiments can utilize a software-based implementation of the functionality described. Certain embodiments combine both types of program code.

FIG. **1** illustrates a system **100** and environment thereof, according to at least one embodiment, by which a user **110** benefits through use of services and products of an enterprise system **200**. The environment may include, for example, a distributed cloud computing environment (private cloud, public cloud, community cloud, and/or hybrid cloud), an on-premise environment, fog computing environment, and/or an edge computing environment. The user **110** accesses services and products by use of one or more user devices, illustrated in separate examples as a computing device **104** and a mobile device **106**, which may be, as non-limiting examples, a smart phone, a portable digital assistant (PDA), a pager, a mobile television, a gaming device, a laptop computer, a camera, a video recorder, an audio/video player, radio, a GPS device, or any combination of the aforementioned, or other portable device with processing and communication capabilities. In the illustrated example, the mobile device **106** is illustrated in FIG. **1** as having exemplary elements, the below descriptions of which apply as well to the computing device **104**, which can be, as non-limiting examples, a desktop computer, a laptop computer, or other user-accessible computing device.

Furthermore, the user device, referring to either or both of the computing device **104** and the mobile device **106**, may be or include a workstation, a server, or any other suitable device, including a set of servers, a cloud-based application or system, or any other suitable system, adapted to execute, for example any suitable operating system, including Linux, UNIX, Windows, macOS, iOS, Android and any other known operating system used on personal computers, central computing systems, phones, and other devices.

The user **110** can be an individual, a group, or any entity in possession of or having access to the user device, referring to either or both of the mobile device **104** and computing device **106**, which may be personal or public items. Although the user **110** may be singly represented in some drawings, at least in some embodiments according to these descriptions the user **110** is one of many such that a market or community of users, consumers, customers, business entities, government entities, clubs, and groups of any size are all within the scope of these descriptions.

The user device, as illustrated with reference to the mobile device **106**, includes components such as, at least one of each of a processing device **120**, and a memory device **122** for processing use, such as random access memory (RAM), and read-only memory (ROM). The illustrated mobile device **106** further includes a storage device **124** including at least one of a non-transitory storage medium, such as a microdrive, for long-term, intermediate-term, and short-term storage of computer-readable instructions **126** for execution by the processing device **120**. For example, the instructions **126** can include instructions for an operating system and various applications or programs **130**, of which the application **132** is represented as a particular example. The storage device **124** can store various other data items **134**, which can include, as non-limiting examples, cached data, user files such as those for pictures, audio and/or video recordings, files downloaded or received from other devices, and other data items preferred by the user or required or related to any or all of the applications or programs **130**.

The memory device **122** is operatively coupled to the processing device **120**. As used herein, memory includes any computer readable medium to store data, code, or other information. The memory device **122** may include volatile memory, such as volatile Random Access Memory (RAM) including a cache area for the temporary storage of data. The memory device **122** may also include non-volatile memory, which can be embedded and/or may be removable. The non-volatile memory can additionally or alternatively include an electrically erasable programmable read-only memory (EEPROM), flash memory or the like.

According to various embodiments, the memory device **122** and storage device **124** may be combined into a single storage medium. The memory device **122** and storage device **124** can store any of a number of applications which comprise computer-executable instructions and code executed by the processing device **120** to implement the functions of the mobile device **106** described herein. For example, the memory device **122** may include such applications as a conventional web browser application and/or a

mobile P2P payment system client application. These applications also typically provide a graphical user interface (GUI) on the display **140** that allows the user **110** to communicate with the mobile device **106**, and, for example a mobile banking system, and/or other devices or systems. In one embodiment, when the user **110** decides to enroll in a mobile banking program, the user **110** downloads or otherwise obtains the mobile banking system client application from a mobile banking system, for example enterprise system **200**, or from a distinct application server. In other embodiments, the user **110** interacts with a mobile banking system via a web browser application in addition to, or instead of, the mobile P2P payment system client application.

The processing device **120**, and other processors described herein, generally include circuitry for implementing communication and/or logic functions of the mobile device **106**. For example, the processing device **120** may include a digital signal processor, a microprocessor, and various analog to digital converters, digital to analog converters, and/or other support circuits. Control and signal processing functions of the mobile device **106** are allocated between these devices according to their respective capabilities. The processing device **120** thus may also include the functionality to encode and interleave messages and data prior to modulation and transmission. The processing device **120** can additionally include an internal data modem. Further, the processing device **120** may include functionality to operate one or more software programs, which may be stored in the memory device **122**, or in the storage device **124**. For example, the processing device **120** may be capable of operating a connectivity program, such as a web browser application. The web browser application may then allow the mobile device **106** to transmit and receive web content, such as, for example, location-based content and/or other web page content, according to a Wireless Application Protocol (WAP), Hypertext Transfer Protocol (HTTP), and/or the like.

The memory device **122** and storage device **124** can each also store any of a number of pieces of information, and data, used by the user device and the applications and devices that facilitate functions of the user device, or are in communication with the user device, to implement the functions described herein and others not expressly described. For example, the storage device may include such data as user authentication information, etc.

The processing device **120**, in various examples, can operatively perform calculations, can process instructions for execution, and can manipulate information. The processing device **120** can execute machine-executable instructions stored in the storage device **124** and/or memory device **122** to thereby perform methods and functions as described or implied herein, for example by one or more corresponding flow charts expressly provided or implied as would be understood by one of ordinary skill in the art to which the subject matters of these descriptions pertain. The processing device **120** can be or can include, as non-limiting examples, a central processing unit (CPU), a microprocessor, a graphics processing unit (GPU), a microcontroller, an application-specific integrated circuit (ASIC), a programmable logic device (PLD), a digital signal processor (DSP), a field programmable gate array (FPGA), a state machine, a controller, gated or transistor logic, discrete physical hardware components, and combinations thereof. In some embodiments, particular portions or steps of methods and functions described herein are performed in whole or in part by way of the processing device **120**, while in other embodiments methods and functions described herein include cloud-based computing in whole or in part such that the processing device **120** facilitates local operations including, as non-limiting examples, communication, data transfer, and user inputs and outputs such as receiving commands from and providing displays to the user.

The mobile device **106**, as illustrated, includes an input and output system **136**, referring to, including, or operatively coupled with, one or more user input devices and/or one or more user output devices, which are operatively coupled to the processing device **120**. The input and output system **136** may include input/output circuitry that may operatively convert analog signals and other signals into digital data or may convert digital data to another type of signal. For example, the input/output circuitry may receive and convert physical contact inputs, physical movements, or auditory signals (e.g., which may be used to authenticate a user) to digital data. Once converted, the digital data may be provided to the processing device **120**. The input and output system **136** may also include a display **140** (e.g., a liquid crystal display (LCD), light emitting diode (LED) display, or the like), which can be, as a non-limiting example, a presence-sensitive input screen (e.g., touch screen or the like) of the mobile device **106**, which serves both as an output device, by providing graphical and text indicia and presentations for viewing by one or more user **110**, and as an input device, by providing virtual buttons, selectable options, a virtual keyboard, and other indicia that, when touched, control the mobile device **106** by user action. The user output devices include a speaker **144** or other audio device. The user input devices, which allow the mobile device **106** to receive data and actions such as button manipulations and touches from a user such as the user **110**, may include any of a number of devices allowing the mobile device **106** to receive data from a user, such as a keypad, keyboard, touch-screen, touchpad, microphone **142**, mouse, joystick, other pointer device, button, soft key, infrared sensor, and/or other input device(s). The input and output system **136** may also include a camera **146**, such as a digital camera.

Further non-limiting examples of input devices and/or output devices include, one or more of each, any, and all of a wireless or wired keyboard, a mouse, a touchpad, a button, a switch, a light, an LED, a buzzer, a bell, a printer and/or other user input devices and output devices for use by or communication with the user **110** in accessing, using, and controlling, in whole or in part, the user device, referring to either or both of the computing device **104** and a mobile device **106**. Inputs by one or more user **110** can thus be made via voice, text or graphical indicia selections. For example, such inputs in some examples correspond to user-side actions and communications seeking services and products of the enterprise system **200**, and at least some outputs in such examples correspond to data representing enterprise-side actions and communications in two-way communications between a user **110** and an enterprise system **200**.

The input and output system **136** may also be configured to obtain and process various forms of authentication via an authentication system to obtain authentication information of a user **110**. Various authentication systems may include, according to various embodiments, a recognition system that detects biometric features or attributes of a user such as, for example fingerprint recognition systems and the like (hand print recognition systems, palm print recognition systems, etc.), iris recognition and the like used to authenticate a user based on features of the user's eyes, facial recognition systems based on facial features of the user, DNA-based

authentication, or any other suitable biometric attribute or information associated with a user. Additionally or alternatively, voice biometric systems may be used to authenticate a user using speech recognition associated with a word, phrase, tone, or other voice-related features of the user. Alternate authentication systems may include one or more systems to identify a user based on a visual or temporal pattern of inputs provided by the user. For instance, the user device may display, for example, selectable options, shapes, inputs, buttons, numeric representations, etc. that must be selected in a pre-determined specified order or according to a specific pattern. Other authentication processes are also contemplated herein including, for example, email authentication, password protected authentication, device verification of saved devices, code-generated authentication, text message authentication, phone call authentication, etc. The user device may enable users to input any number or combination of authentication systems.

The user device, referring to either or both of the computing device **104** and the mobile device **106** may also include a positioning device **108**, which can be for example a global positioning system device (GPS) configured to be used by a positioning system to determine a location of the computing device **104** or mobile device **106**. For example, the positioning system device **108** may include a GPS transceiver. In some embodiments, the positioning system device **108** includes an antenna, transmitter, and receiver. For example, in one embodiment, triangulation of cellular signals may be used to identify the approximate location of the mobile device **106**. In other embodiments, the positioning device **108** includes a proximity sensor or transmitter, such as an RFID tag, which can sense or be sensed by devices known to be located proximate a merchant or other location to determine that the consumer mobile device **106** is located proximate these known devices.

In the illustrated example, a system intraconnect **138**, connects, for example electrically, the various described, illustrated, and implied components of the mobile device **106**. The intraconnect **138**, in various non-limiting examples, can include or represent, a system bus, a high-speed interface connecting the processing device **120** to the memory device **122**, individual electrical connections among the components, and electrical conductive traces on a motherboard common to some or all of the above-described components of the user device (referring to either or both of the computing device **104** and the mobile device **106**). As discussed herein, the system intraconnect **138** may operatively couple various components with one another, or in other words, electrically connects those components, either directly or indirectly—by way of intermediate component(s)—with one another.

The user device, referring to either or both of the computing device **104** and the mobile device **106**, with particular reference to the mobile device **106** for illustration purposes, includes a communication interface **150**, by which the mobile device **106** communicates and conducts transactions with other devices and systems. The communication interface **150** may include digital signal processing circuitry and may provide two-way communications and data exchanges, for example wirelessly via wireless communication device **152**, and for an additional or alternative example, via wired or docked communication by mechanical electrically conductive connector **154**. Communications may be conducted via various modes or protocols, of which GSM voice calls, SMS, EMS, MMS messaging, TDMA, CDMA, PDC, WCDMA, CDMA2000, and GPRS, are all non-limiting and non-exclusive examples. Thus, communications can be conducted, for example, via the wireless communication device **152**, which can be or include a radio-frequency transceiver, a Bluetooth device, Wi-Fi device, a Near-field communication device, and other transceivers. In addition, GPS (Global Positioning System) may be included for navigation and location-related data exchanges, ingoing and/or outgoing. Communications may also or alternatively be conducted via the connector **154** for wired connections such by USB, Ethernet, and other physically connected modes of data transfer.

The processing device **120** is configured to use the communication interface **150** as, for example, a network interface to communicate with one or more other devices on a network. In this regard, the communication interface **150** utilizes the wireless communication device **152** as an antenna operatively coupled to a transmitter and a receiver (together a "transceiver") included with the communication interface **150**. The processing device **120** is configured to provide signals to and receive signals from the transmitter and receiver, respectively. The signals may include signaling information in accordance with the air interface standard of the applicable cellular system of a wireless telephone network. In this regard, the mobile device **106** may be configured to operate with one or more air interface standards, communication protocols, modulation types, and access types. By way of illustration, the mobile device **106** may be configured to operate in accordance with any of a number of first, second, third, fourth, fifth-generation communication protocols and/or the like. For example, the mobile device **106** may be configured to operate in accordance with second-generation (2G) wireless communication protocols IS-136 (time division multiple access (TDMA)), GSM (global system for mobile communication), and/or IS-95 (code division multiple access (CDMA)), or with third-generation (3G) wireless communication protocols, such as Universal Mobile Telecommunications System (UMTS), CDMA2000, wideband CDMA (WCDMA) and/or time division-synchronous CDMA (TD-SCDMA), with fourth-generation (4G) wireless communication protocols such as Long-Term Evolution (LTE), fifth-generation (5G) wireless communication protocols, Bluetooth Low Energy (BLE) communication protocols such as Bluetooth 5.0, ultra-wideband (UWB) communication protocols, and/or the like. The mobile device **106** may also be configured to operate in accordance with non-cellular communication mechanisms, such as via a wireless local area network (WLAN) or other communication/data networks.

The communication interface **150** may also include a payment network interface. The payment network interface may include software, such as encryption software, and hardware, such as a modem, for communicating information to and/or from one or more devices on a network. For example, the mobile device **106** may be configured so that it can be used as a credit or debit card by, for example, wirelessly communicating account numbers or other authentication information to a terminal of the network. Such communication could be performed via transmission over a wireless communication protocol such as the Near-field communication protocol.

The mobile device **106** further includes a power source **128**, such as a battery, for powering various circuits and other devices that are used to operate the mobile device **106**. Embodiments of the mobile device **106** may also include a clock or other timer configured to determine and, in some cases, communicate actual or relative time to the processing device **120** or one or more other devices. For further example, the clock may facilitate timestamping transmis-

sions, receptions, and other data for security, authentication, logging, polling, data expiry, and forensic purposes.

System **100** as illustrated diagrammatically represents at least one example of a possible implementation, where alternatives, additions, and modifications are possible for performing some or all of the described methods, operations and functions. Although shown separately, in some embodiments, two or more systems, servers, or illustrated components may utilized. In some implementations, the functions of one or more systems, servers, or illustrated components may be provided by a single system or server. In some embodiments, the functions of one illustrated system or server may be provided by multiple systems, servers, or computing devices, including those physically located at a central facility, those logically local, and those located as remote with respect to each other.

The enterprise system **200** can offer any number or type of services and products to one or more users **110**. In some examples, an enterprise system **200** offers products. In some examples, an enterprise system **200** offers services. Use of "service(s)" or "product(s)" thus relates to either or both in these descriptions. With regard, for example, to online information and financial services, "service" and "product" are sometimes termed interchangeably. In non-limiting examples, services and products include retail services and products, information services and products, custom services and products, predefined or pre-offered services and products, consulting services and products, advising services and products, forecasting services and products, internet products and services, social media, and financial services and products, which may include, in non-limiting examples, services and products relating to banking, checking, savings, investments, credit cards, automatic-teller machines, debit cards, loans, mortgages, personal accounts, business accounts, account management, credit reporting, credit requests, and credit scores.

To provide access to, or information regarding, some or all the services and products of the enterprise system **200**, automated assistance may be provided by the enterprise system **200**. For example, automated access to user accounts and replies to inquiries may be provided by enterprise-side automated voice, text, and graphical display communications and interactions. In at least some examples, any number of human agents **210**, can be employed, utilized, authorized or referred by the enterprise system **200**. Such human agents **210** can be, as non-limiting examples, point of sale or point of service (POS) representatives, online customer service assistants available to users **110**, advisors, managers, sales team members, and referral agents ready to route user requests and communications to preferred or particular other agents, human or virtual.

Human agents **210** may utilize agent devices **212** to serve users in their interactions to communicate and take action. The agent devices **212** can be, as non-limiting examples, computing devices, kiosks, terminals, smart devices such as phones, and devices and tools at customer service counters and windows at POS locations. In at least one example, the diagrammatic representation of the components of the user device **106** in FIG. **1** applies as well to one or both of the computing device **104** and the agent devices **212**.

Agent devices **212** individually or collectively include input devices and output devices, including, as non-limiting examples, a touch screen, which serves both as an output device by providing graphical and text indicia and presentations for viewing by one or more agent **210**, and as an input device by providing virtual buttons, selectable options, a virtual keyboard, and other indicia that, when touched or activated, control or prompt the agent device **212** by action of the attendant agent **210**. Further non-limiting examples include, one or more of each, any, and all of a keyboard, a mouse, a touchpad, a joystick, a button, a switch, a light, an LED, a microphone serving as input device for example for voice input by a human agent **210**, a speaker serving as an output device, a camera serving as an input device, a buzzer, a bell, a printer and/or other user input devices and output devices for use by or communication with a human agent **210** in accessing, using, and controlling, in whole or in part, the agent device **212**.

Inputs by one or more human agents **210** can thus be made via voice, text or graphical indicia selections. For example, some inputs received by an agent device **212** in some examples correspond to, control, or prompt enterprise-side actions and communications offering services and products of the enterprise system **200**, information thereof, or access thereto. At least some outputs by an agent device **212** in some examples correspond to, or are prompted by, user-side actions and communications in two-way communications between a user **110** and an enterprise-side human agent **210**.

From a user perspective experience, an interaction in some examples within the scope of these descriptions begins with direct or first access to one or more human agents **210** in person, by phone, or online for example via a chat session or website function or feature. In other examples, a user is first assisted by a virtual agent **214** of the enterprise system **200**, which may satisfy user requests or prompts by voice, text, or online functions, and may refer users to one or more human agents **210** once preliminary determinations or conditions are made or met.

A computing system **206** of the enterprise system **200** may include components such as, at least one of each of a processing device **220**, and a memory device **222** for processing use, such as random access memory (RAM), and read-only memory (ROM). The illustrated computing system **206** further includes a storage device **224** including at least one non-transitory storage medium, such as a microdrive, for long-term, intermediate-term, and short-term storage of computer-readable instructions **226** for execution by the processing device **220**. For example, the instructions **226** can include instructions for an operating system and various applications or programs **230**, of which the application **232** is represented as a particular example. The storage device **224** can store various other data **234**, which can include, as non-limiting examples, cached data, and files such as those for user accounts, user profiles, account balances, and transaction histories, files downloaded or received from other devices, and other data items preferred by the user or required or related to any or all of the applications or programs **230**.

The computing system **206**, in the illustrated example, includes an input/output system **236**, referring to, including, or operatively coupled with input devices and output devices such as, in a non-limiting example, agent devices **212**, which have both input and output capabilities.

In the illustrated example, a system intraconnect **238** electrically connects the various above-described components of the computing system **206**. In some cases, the intraconnect **238** operatively couples components to one another, which indicates that the components may be directly or indirectly connected, such as by way of one or more intermediate components. The intraconnect **238**, in various non-limiting examples, can include or represent, a system bus, a high-speed interface connecting the processing device **220** to the memory device **222**, individual electrical connections among the components, and electrical

conductive traces on a motherboard common to some or all of the above-described components of the user device.

The computing system **206**, in the illustrated example, includes a communication interface **250**, by which the computing system **206** communicates and conducts transactions with other devices and systems. The communication interface **250** may include digital signal processing circuitry and may provide two-way communications and data exchanges, for example wirelessly via wireless device **252**, and for an additional or alternative example, via wired or docked communication by mechanical electrically conductive connector **254**. Communications may be conducted via various modes or protocols, of which GSM voice calls, SMS, EMS, MMS messaging, TDMA, CDMA, PDC, WCDMA, CDMA2000, and GPRS, are all non-limiting and non-exclusive examples. Thus, communications can be conducted, for example, via the wireless device **252**, which can be or include a radio-frequency transceiver, a Bluetooth device, Wi-Fi device, Near-field communication device, and other transceivers. In addition, GPS (Global Positioning System) may be included for navigation and location-related data exchanges, ingoing and/or outgoing. Communications may also or alternatively be conducted via the connector **254** for wired connections such as by USB, Ethernet, and other physically connected modes of data transfer.

The processing device **220**, in various examples, can operatively perform calculations, can process instructions for execution, and can manipulate information. The processing device **220** can execute machine-executable instructions stored in the storage device **224** and/or memory device **222** to thereby perform methods and functions as described or implied herein, for example by one or more corresponding flow charts expressly provided or implied as would be understood by one of ordinary skill in the art to which the subjects matters of these descriptions pertain. The processing device **220** can be or can include, as non-limiting examples, a central processing unit (CPU), a microprocessor, a graphics processing unit (GPU), a microcontroller, an application-specific integrated circuit (ASIC), a programmable logic device (PLD), a digital signal processor (DSP), a field programmable gate array (FPGA), a state machine, a controller, gated or transistor logic, discrete physical hardware components, and combinations thereof.

Furthermore, the computing system **206**, may be or include a workstation, a server, or any other suitable device, including a set of servers, a cloud-based application or system, or any other suitable system, adapted to execute, for example any suitable operating system, including Linux, UNIX, Windows, macOS, iOS, Android, and any known other operating system used on personal computer, central computing systems, phones, and other devices.

The user devices, referring to either or both of the computing device **104** and mobile device **106**, the agent devices **212**, and the enterprise computing system **206**, which may be one or any number centrally located or distributed, are in communication through one or more networks, referenced as network **258** in FIG. **1**.

Network **258** provides wireless or wired communications among the components of the system **100** and the environment thereof, including other devices local or remote to those illustrated, such as additional mobile devices, servers, and other devices communicatively coupled to network **258**, including those not illustrated in FIG. **1**. The network **258** is singly depicted for illustrative convenience but may include more than one network without departing from the scope of these descriptions. In some embodiments, the network **258** may be or provide one or more cloud-based services or operations. The network **258** may be or include an enterprise or secured network, or may be implemented, at least in part, through one or more connections to the Internet. A portion of the network **258** may be a virtual private network (VPN) or an Intranet. The network **258** can include wired and wireless links, including, as non-limiting examples, 802.11a/b/g/n/ac, 802.20, WiMax, LTE, and/or any other wireless link. The network **258** may include any internal or external network, networks, sub-network, and combinations of such operable to implement communications between various computing components within and beyond the illustrated system **100**. The network **258** may communicate, for example, Internet Protocol (IP) packets, Frame Relay frames, Asynchronous Transfer Mode (ATM) cells, voice, video, data, and other suitable information between network addresses. The network **258** may also include one or more local area networks (LANs), radio access networks (RANs), metropolitan area networks (MANs), wide area networks (WANs), all or a portion of the internet and/or any other communication system or systems at one or more locations.

The network **258** may incorporate a cloud platform/data center that support various service models including Platform as a Service (PaaS), Infrastructure-as-a-Service (IaaS), and Software-as-a-Service (SaaS). Such service models may provide, for example, a digital platform accessible to the user device (referring to either or both of the computing device **104** and the mobile device **106**). Specifically, SaaS may provide a user with the capability to use applications running on a cloud infrastructure, where the applications are accessible via a thin client interface such as a web browser and the user is not permitted to manage or control the underlying cloud infrastructure (i.e., network, servers, operating systems, storage, or specific application capabilities that are not user-specific). PaaS also do not permit the user to manage or control the underlying cloud infrastructure, but this service may enable a user to deploy user-created or acquired applications onto the cloud infrastructure using programming languages and tools provided by the provider of the application. In contrast, IaaS provides a user the permission to provision processing, storage, networks, and other computing resources as well as run arbitrary software (e.g., operating systems and applications) thereby giving the user control over operating systems, storage, deployed applications, and potentially select networking components (e.g., host firewalls).

The network **258** may also incorporate various cloud-based deployment models including private cloud (i.e., an organization-based cloud managed by either the organization or third parties and hosted on-premises or off premises), public cloud (i.e., cloud-based infrastructure available to the general public that is owned by an organization that sells cloud services), community cloud (i.e., cloud-based infrastructure shared by several organizations and manages by the organizations or third parties and hosted on-premises or off premises), and/or hybrid cloud (i.e., composed of two or more clouds e.g., private community, and/or public).

Two external systems **202** and **204** are expressly illustrated in FIG. **1**, representing any number and variety of data sources, users, consumers, customers, business entities, banking systems, government entities, clubs, and groups of any size are all within the scope of the descriptions. In at least one example, the external systems **202** and **204** represent automatic teller machines (ATMs) utilized by the enterprise system **200** in serving users **110**. In another example, the external systems **202** and **204** represent payment clearinghouse or payment rail systems for processing payment transactions, and in another example, the external systems

202 and 204 represent third party systems such as merchant systems configured to interact with the user device 106 during transactions and also configured to interact with the enterprise system 200 in back-end transactions clearing processes.

In certain embodiments, one or more of the systems such as the user device (referring to either or both of the computing device 104 and the mobile device 106), the enterprise system 200, and/or the external systems 202 and 204 are, include, or utilize virtual resources. In some cases, such virtual resources are considered cloud resources or virtual machines. The cloud computing configuration may provide an infrastructure that includes a network of interconnected nodes and provides stateless, low coupling, modularity, and semantic interoperability. Such interconnected nodes may incorporate a computer system that includes one or more processors, a memory, and a bus that couples various system components (e.g., the memory) to the processor. Such virtual resources may be available for shared use among multiple distinct resource consumers and in certain implementations, virtual resources do not necessarily correspond to one or more specific pieces of hardware, but rather to a collection of pieces of hardware operatively coupled within a cloud computing configuration so that the resources may be shared as needed.

As used herein, an artificial intelligence system, artificial intelligence algorithm, artificial intelligence module, program, and the like, generally refer to computer implemented programs that are suitable to simulate intelligent behavior (i.e., intelligent human behavior) and/or computer systems and associated programs suitable to perform tasks that typically require a human to perform, such as tasks requiring visual perception, speech recognition, decision-making, translation, and the like. An artificial intelligence system may include, for example, at least one of a series of associated if-then logic statements, a statistical model suitable to map raw sensory data into symbolic categories and the like, or a machine learning program. A machine learning program, machine learning algorithm, or machine learning module, as used herein, is generally a type of artificial intelligence including one or more algorithms that can learn and/or adjust parameters based on input data provided to the algorithm. In some instances, machine learning programs, algorithms, and modules are used at least in part in implementing artificial intelligence (AI) functions, systems, and methods.

Artificial Intelligence and/or machine learning programs may be associated with or conducted by one or more processors, memory devices, and/or storage devices of a computing system or device. It should be appreciated that the AI algorithm or program may be incorporated within the existing system architecture or be configured as a standalone modular component, controller, or the like communicatively coupled to the system. An AI program and/or machine learning program may generally be configured to perform methods and functions as described or implied herein, for example by one or more corresponding flow charts expressly provided or implied as would be understood by one of ordinary skill in the art to which the subjects matters of these descriptions pertain.

A machine learning program may be configured to use various analytical tools (e.g., algorithmic applications) to leverage data to make predictions or decisions. Machine learning programs may be configured to implement various algorithmic processes and learning approaches including, for example, decision tree learning, association rule learning, artificial neural networks, recurrent artificial neural networks, long short term memory networks, inductive logic programming, support vector machines, clustering, Bayesian networks, reinforcement learning, representation learning, similarity and metric learning, sparse dictionary learning, genetic algorithms, k-nearest neighbor (KNN), and the like. In some embodiments, the machine learning algorithm may include one or more image recognition algorithms suitable to determine one or more categories to which an input, such as data communicated from a visual sensor or a file in JPEG, PNG or other format, representing an image or portion thereof, belongs. Additionally, or alternatively, the machine learning algorithm may include one or more regression algorithms configured to output a numerical value given an input. Further, the machine learning may include one or more pattern recognition algorithms, e.g., a module, subroutine or the like capable of translating text or string characters and/or a speech recognition module or subroutine. In various embodiments, the machine learning module may include a machine learning acceleration logic, e.g., a fixed function matrix multiplication logic, in order to implement the stored processes and/or optimize the machine learning logic training and interface.

Machine learning models are trained using various data inputs and techniques. Example training methods may include, for example, supervised learning, (e.g., decision tree learning, support vector machines, similarity and metric learning, etc.), unsupervised learning, (e.g., association rule learning, clustering, etc.), reinforcement learning, semi-supervised learning, self-supervised learning, multi-instance learning, inductive learning, deductive inference, transductive learning, sparse dictionary learning and the like. Example clustering algorithms used in unsupervised learning may include, for example, k-means clustering, density based special clustering of applications with noise (DBSCAN), mean shift clustering, expectation maximization (EM) clustering using Gaussian mixture models (GMM), agglomerative hierarchical clustering, or the like. According to one embodiment, clustering of data may be performed using a cluster model to group data points based on certain similarities using unlabeled data. Example cluster models may include, for example, connectivity models, centroid models, distribution models, density models, group models, graph-based models, neural models and the like.

One subfield of machine learning includes neural networks, which take inspiration from biological neural networks. In machine learning, a neural network includes interconnected units that process information by responding to external inputs to find connections and derive meaning from undefined data. A neural network can, in a sense, learn to perform tasks by interpreting numerical patterns that take the shape of vectors and by categorizing data based on similarities, without being programmed with any task-specific rules. A neural network generally includes connected units, neurons, or nodes (e.g., connected by synapses) and may allow for the machine learning program to improve performance. A neural network may define a network of functions, which have a graphical relationship. Various neural networks that implement machine learning exist including, for example, feedforward artificial neural networks, perceptron and multilayer perceptron neural networks, radial basis function artificial neural networks, recurrent artificial neural networks, modular neural networks, long short-term memory networks, as well as various other neural networks.

Neural networks may perform a supervised learning process where known inputs and known outputs are utilized to categorize, classify, or predict a quality of a future input.

However, additional or alternative embodiments of the machine learning program may be trained utilizing unsupervised or semi-supervised training, where none of the outputs or some of the outputs are unknown, respectively. Typically, a machine learning algorithm is trained (e.g., utilizing a training data set) prior to modeling the problem with which the algorithm is associated. Supervised training of the neural network may include choosing a network topology suitable for the problem being modeled by the network and providing a set of training data representative of the problem. Generally, the machine learning algorithm may adjust the weight coefficients until any error in the output data generated by the algorithm is less than a predetermined, acceptable level. For instance, the training process may include comparing the generated output produced by the network in response to the training data with a desired or correct output. An associated error amount may then be determined for the generated output data, such as for each output data point generated in the output layer. The associated error amount may be communicated back through the system as an error signal, where the weight coefficients assigned in the hidden layer are adjusted based on the error signal. For instance, the associated error amount (e.g., a value between −1 and 1) may be used to modify the previous coefficient, e.g., a propagated value. The machine learning algorithm may be considered sufficiently trained when the associated error amount for the output data is less than the predetermined, acceptable level (e.g., each data point within the output layer includes an error amount less than the predetermined, acceptable level). Thus, the parameters determined from the training process can be utilized with new input data to categorize, classify, and/or predict other values based on the new input data.

An artificial neural network (ANN), also known as a feedforward network, may be utilized, e.g., an acyclic graph with nodes arranged in layers. A feedforward network (see, e.g., feedforward network **260** referenced in FIG. **2**A) may include a topography with a hidden layer **264** between an input layer **262** and an output layer **266**. The input layer **262**, having nodes commonly referenced in FIG. **2**A as input nodes **272** for convenience, communicates input data, variables, matrices, or the like to the hidden layer **264**, having nodes **274**. The hidden layer **264** generates a representation and/or transformation of the input data into a form that is suitable for generating output data. Adjacent layers of the topography are connected at the edges of the nodes of the respective layers, but nodes within a layer typically are not separated by an edge. In at least one embodiment of such a feedforward network, data is communicated to the nodes **272** of the input layer, which then communicates the data to the hidden layer **264**. The hidden layer **264** may be configured to determine the state of the nodes in the respective layers and assign weight coefficients or parameters of the nodes based on the edges separating each of the layers, e.g., an activation function implemented between the input data communicated from the input layer **262** and the output data communicated to the nodes **276** of the output layer **266**. It should be appreciated that the form of the output from the neural network may generally depend on the type of model represented by the algorithm. Although the feedforward network **260** of FIG. **2**A expressly includes a single hidden layer **264**, other embodiments of feedforward networks within the scope of the descriptions can include any number of hidden layers. The hidden layers are intermediate the input and output layers and are generally where all or most of the computation is done.

An additional or alternative type of neural network suitable for use in the machine learning program and/or module is a Convolutional Neural Network (CNN). A CNN is a type of feedforward neural network that may be utilized to model data associated with input data having a grid-like topology. In some embodiments, at least one layer of a CNN may include a sparsely connected layer, in which each output of a first hidden layer does not interact with each input of the next hidden layer. For example, the output of the convolution in the first hidden layer may be an input of the next hidden layer, rather than a respective state of each node of the first layer. CNNs are typically trained for pattern recognition, such as speech processing, language processing, and visual processing. As such, CNNs may be particularly useful for implementing optical and pattern recognition programs required from the machine learning program. A CNN includes an input layer, a hidden layer, and an output layer, typical of feedforward networks, but the nodes of a CNN input layer are generally organized into a set of categories via feature detectors and based on the receptive fields of the sensor, retina, input layer, etc. Each filter may then output data from its respective nodes to corresponding nodes of a subsequent layer of the network. A CNN may be configured to apply the convolution mathematical operation to the respective nodes of each filter and communicate the same to the corresponding node of the next subsequent layer. As an example, the input to the convolution layer may be a multidimensional array of data. The convolution layer, or hidden layer, may be a multidimensional array of parameters determined while training the model.

An exemplary convolutional neural network CNN is depicted and referenced as **280** in FIG. **2**B. As in the basic feedforward network **260** of FIG. **2**A, the illustrated example of FIG. **2**B has an input layer **282** and an output layer **286**. However, where a single hidden layer **264** is represented in FIG. **2**A, multiple consecutive hidden layers **284**A, **284**B, and **284**C are represented in FIG. **2**B. The edge neurons represented by white-filled arrows highlight that hidden layer nodes can be connected locally, such that not all nodes of succeeding layers are connected by neurons. FIG. **2**C, representing a portion of the convolutional neural network **280** of FIG. **2**B, specifically portions of the input layer **282** and the first hidden layer **284**A, illustrates that connections can be weighted. In the illustrated example, labels **W1** and **W2** refer to respective assigned weights for the referenced connections. Two hidden nodes **283** and **285** share the same set of weights **W1** and **W2** when connecting to two local patches.

Weight defines the impact a node in any given layer has on computations by a connected node in the next layer. FIG. **3** represents a particular node **300** in a hidden layer. The node **300** is connected to several nodes in the previous layer representing inputs to the node **300**. The input nodes **301**, **302**, **303** and **304** are each assigned a respective weight **W01**, **W02**, **W03**, and **W04** in the computation at the node **300**, which in this example is a weighted sum.

An additional or alternative type of feedforward neural network suitable for use in the machine learning program and/or module is a Recurrent Neural Network (RNN). An RNN may allow for analysis of sequences of inputs rather than only considering the current input data set. RNNs typically include feedback loops/connections between layers of the topography, thus allowing parameter data to be communicated between different parts of the neural network. RNNs typically have an architecture including cycles, where past values of a parameter influence the current calculation of the parameter, e.g., at least a portion of the

output data from the RNN may be used as feedback/input in calculating subsequent output data. In some embodiments, the machine learning module may include an RNN configured for language processing, e.g., an RNN configured to perform statistical language modeling to predict the next word in a string based on the previous words. The RNN(s) of the machine learning program may include a feedback system suitable to provide the connection(s) between subsequent and previous layers of the network.

An example for a Recurrent Neural Network RNN is referenced as **400** in FIG. **4**. As in the basic feedforward network **260** of FIG. **2**A, the illustrated example of FIG. **4** has an input layer **410** (with nodes **412**) and an output layer **440** (with nodes **442**). However, where a single hidden layer **264** is represented in FIG. **2**A, multiple consecutive hidden layers **420** and **430** are represented in FIG. **4** (with nodes **422** and nodes **432**, respectively). As shown, the RNN **400** includes a feedback connector **404** configured to communicate parameter data from at least one node **432** from the second hidden layer **430** to at least one node **422** of the first hidden layer **420**. It should be appreciated that two or more and up to all of the nodes of a subsequent layer may provide or communicate a parameter or other data to a previous layer of the RNN **400**. Moreover, and in some embodiments, the RNN **400** may include multiple feedback connectors **404** (e.g., connectors **404** suitable to communicatively couple pairs of nodes and/or connector systems **404** configured to provide communication between three or more nodes). Additionally, or alternatively, the feedback connector **404** may communicatively couple two or more nodes having at least one hidden layer between them, i.e., nodes of nonsequential layers of the RNN **400**.

In an additional or alternative embodiment, the machine-learning program may include one or more support vector machines. A support vector machine may be configured to determine a category to which input data belongs. For example, the machine-learning program may be configured to define a margin using a combination of two or more of the input variables and/or data points as support vectors to maximize the determined margin. Such a margin may generally correspond to a distance between the closest vectors that are classified differently. The machine-learning program may be configured to utilize a plurality of support vector machines to perform a single classification. For example, the machine-learning program may determine the category to which input data belongs using a first support vector determined from first and second data points/variables, and the machine-learning program may independently categorize the input data using a second support vector determined from third and fourth data points/variables. The support vector machine(s) may be trained similarly to the training of neural networks, e.g., by providing a known input vector (including values for the input variables) and a known output classification. The support vector machine is trained by selecting the support vectors and/or a portion of the input vectors that maximize the determined margin.

As depicted, and in some embodiments, the machine-learning program may include a neural network topography having more than one hidden layer. In such embodiments, one or more of the hidden layers may have a different number of nodes and/or the connections defined between layers. In some embodiments, each hidden layer may be configured to perform a different function. As an example, a first layer of the neural network may be configured to reduce a dimensionality of the input data, and a second layer of the neural network may be configured to perform statistical programs on the data communicated from the first layer. In various embodiments, each node of the previous layer of the network may be connected to an associated node of the subsequent layer (dense layers). Generally, the neural network(s) of the machine-learning program may include a relatively large number of layers, e.g., three or more layers, and may be referred to as deep neural networks. For example, the node of each hidden layer of a neural network may be associated with an activation function utilized by the machine-learning program to generate an output received by a corresponding node in the subsequent layer. The last hidden layer of the neural network communicates a data set (e.g., the result of data processed within the respective layer) to the output layer. Deep neural networks may require more computational time and power to train, but the additional hidden layers provide multistep pattern recognition capability and/or reduced output error relative to simple or shallow machine learning architectures (e.g., including only one or two hidden layers).

According to various implementations, deep neural networks incorporate neurons, synapses, weights, biases, and functions and can be trained to model complex non-linear relationships. Various deep learning frameworks may include, for example, TensorFlow, MxNet, PyTorch, Keras, Gluon, and the like. Training a deep neural network may include complex input/output transformations and may include, according to various embodiments, a backpropagation algorithm. According to various embodiments, deep neural networks may be configured to classify images of handwritten digits from a dataset or various other images. According to various embodiments, the datasets may include a collection of files that are unstructured and lack predefined data model schema or organization. Unlike structured data, which is usually stored in a relational database (RDBMS) and can be mapped into designated fields, unstructured data comes in many formats that can be challenging to process and analyze. Examples of unstructured data may include, according to non-limiting examples, dates, numbers, facts, emails, text files, scientific data, satellite imagery, media files, social media data, text messages, mobile communication data, and the like.

Referring now to FIG. **5** and some embodiments, an AI program **502** may include a front-end algorithm **504** and a back-end algorithm **506**. The artificial intelligence program **502** may be implemented on an AI processor **520**, such as the processing device **120**, the processing device **220**, and/or a dedicated processing device. The instructions associated with the front-end algorithm **504** and the back-end algorithm **506** may be stored in an associated memory device and/or storage device of the system (e.g., storage device **124**, memory device **122**, storage device **224**, and/or memory device **222**) communicatively coupled to the AI processor **520**, as shown. Additionally, or alternatively, the system may include one or more memory devices and/or storage devices (represented by memory **524** in FIG. **5**) for processing use and/or including one or more instructions necessary for operation of the AI program **502**. In some embodiments, the AI program **502** may include a deep neural network (e.g., a front-end algorithm **504** configured to perform pre-processing, such as feature recognition, and a back-end algorithm **506** configured to perform an operation on the data set communicated directly or indirectly to the back-end algorithm **506** that together form the deep neural network). For instance, the front-end algorithm **504** can include at least one CNN **508** communicatively coupled to send output data to the back-end algorithm **506**.

Additionally, or alternatively, the front-end algorithm **504** can include one or more AI algorithms **510**, **512** (e.g.,

statistical models or machine learning programs such as decision tree learning, associate rule learning, recurrent artificial neural networks, support vector machines, and the like). In various embodiments, the front-end algorithm **504** may be configured to include built in training and inference logic or suitable software to train the neural network prior to use (e.g., machine learning logic including, but not limited to, image recognition, mapping and localization, autonomous navigation, speech synthesis, document imaging, or language translation such as natural language processing). For example, a CNN **508** and/or AI algorithm **510** may be used for image recognition, input categorization, and/or support vector training. In some embodiments and within the front-end algorithm **504**, an output from an AI algorithm **510** may be communicated to a CNN **508** or **509**, which processes the data before communicating an output from the CNN **508**, **509** and/or the front-end algorithm **504** to the back-end algorithm **506**. In various embodiments, the back-end algorithm **506** may be configured to implement input and/or model classification, speech recognition, translation, and the like. For instance, the back-end network **506** may include one or more CNNs (e.g., CNN **514**) or dense networks (e.g., dense networks **516**), as described herein.

For instance, and in some embodiments of the AI program **502**, the program may be configured to perform unsupervised learning, in which the machine learning program performs the training process using unlabeled data, e.g., without known output data with which to compare. During such unsupervised learning, the neural network may be configured to generate groupings of the input data and/or determine how individual input data points are related to the complete input data set (e.g., via the front-end algorithm **504**). For example, unsupervised training may be used to configure a neural network to generate a self-organizing map, reduce the dimensionally of the input data set, and/or to perform outlier/anomaly determinations to identify data points in the data set that falls outside the normal pattern of the data. In some embodiments, the AI program **502** may be trained using a semi-supervised learning process in which some but not all of the output data is known, e.g., a mix of labeled and unlabeled data having the same distribution.

In some embodiments, the AI program **502** may be accelerated via a machine-learning framework **522** (e.g., hardware). The machine learning framework may include an index of basic operations, subroutines, and the like (primitives) typically implemented by AI and/or machine learning algorithms. Thus, the AI program **502** may be configured to utilize the primitives of the framework **522** to perform some or all of the calculations required by the AI program **502**. Primitives suitable for inclusion in the machine learning framework **522** include operations associated with training a convolutional neural network (e.g., pools), tensor convolutions, activation functions, basic algebraic subroutines and programs (e.g., matrix operations, vector operations), numerical method subroutines and programs, and the like.

It should be appreciated that the machine-learning program may include variations, adaptations, and alternatives suitable to perform the operations necessary for the system, and the present disclosure is equally applicable to such suitably configured machine learning and/or artificial intelligence programs, modules, etc. For instance, the machine-learning program may include one or more long short-term memory (LSTM) RNNs, convolutional deep belief networks, deep belief networks DBNs, and the like. DBNs, for instance, may be utilized to pre-train the weighted characteristics and/or parameters using an unsupervised learning process. Further, the machine-learning module may include one or more other machine learning tools (e.g., Logistic Regression (LR), Naive-Bayes, Random Forest (RF), matrix factorization, and support vector machines) in addition to, or as an alternative to, one or more neural networks, as described herein.

FIG. **6** is a flow chart representing a method **600**, according to at least one embodiment, of model development and deployment by machine learning. The method **600** represents at least one example of a machine learning workflow in which steps are implemented in a machine-learning project.

In step **602**, a user authorizes, requests, manages, or initiates the machine-learning workflow. This may represent a user such as human agent, or customer, requesting machine-learning assistance or AI functionality to simulate intelligent behavior (such as a virtual agent) or other machine-assisted or computerized tasks that may, for example, entail visual perception, speech recognition, decision-making, translation, forecasting, predictive modelling, and/or suggestions as non-limiting examples. In a first iteration from the user perspective, step **602** can represent a starting point. However, with regard to continuing or improving an ongoing machine learning workflow, step **602** can represent an opportunity for further user input or oversight via a feedback loop. Such feedback may flow through a user, or in various embodiments, the method automatically provides feedback, retrains and redeploys the retrained model.

In step **604**, data is received, collected, accessed, or otherwise acquired and entered as can be termed data ingestion. In step **606**, the data ingested in step **604** is pre-processed, for example, by cleaning, and/or transformation such as into a format that the following components can digest. The incoming data may be versioned to connect a data snapshot with the particularly resulting trained model. As newly trained models are tied to a set of versioned data, preprocessing steps are tied to the developed model. If new data is subsequently collected and entered, a new model will be generated. If the preprocessing step **606** is updated with newly ingested data, an updated model will be generated. Step **606** can include data validation, which focuses on confirming that the statistics of the ingested data are as expected, such as that data values are within expected numerical ranges, that data sets are within any expected or required categories, and that data comply with any needed distributions such as within those categories. Step **606** can proceed to step **608** to automatically alert the initiating user, other human or virtual agents, and/or other systems, if any anomalies are detected in the data, thereby pausing or terminating the process flow until corrective action is taken.

In step **610**, training test data such as a target variable value is inserted into an iterative training and testing loop. In step **612**, model training, a core step of the machine learning workflow, is implemented. A model architecture is trained in the iterative training and testing loop. For example, features in the training test data are used to train the model based on weights and iterative calculations in which the target variable may be incorrectly predicted in an early iteration as determined by comparison in step **614**, where the model is tested. Subsequent iterations of the model training, in step **612**, may be conducted with updated weights in the calculations. During each iteration of the training and testing loop, the accuracy of the model may be evaluated. In one embodiment, the re-evaluation of the model can include comparing an output of the model with an actual target result or variable to determine the accuracy of the prediction. If the model is not satisfying a minimum

threshold level of accuracy (i.e., the model is underfitted), the system may automatically determine that the threshold level of accuracy is not satisfied and may adjust the weights for a subsequent iteration of the training and testing loop. The weights may be iteratively adjusted during each iteration of the training and testing loop based on the comparison to the threshold level of accuracy. However, there is a balance for training the model in order to avoid overfitting when the model would not perform well on predictions of new data. Rather, the model is automatically trained to be well-fitted such that it satisfies a threshold level of accuracy without learning the noise in the data to the extent that the model would not apply to new data by preventing additional iterations of the training and testing once a maximum accuracy threshold value has been obtained. Thus, with each iteration of the training and testing loop, the accuracy of the model is improved and the iterative training and testing of the model provides an improvement to the performance of a computer and computing technology because the system may automatically determine how many iterations to perform so that the model is well-fitted by surpassing the minimum threshold level of accuracy while automatically stopping the iterative training and testing of the model before the maximum accuracy threshold is obtained. In some embodiments, the training and testing loop utilizes a backpropagation algorithm and a gradient descent algorithm. Gradient descent is an optimization algorithm used to minimize differentiable real-valued multivariate functions. Gradient descent is an optimization algorithm used to minimize differentiable real-valued multivariate functions. The gradient descent algorithm may be used to iteratively adjust model parameters using calculated derivatives to minimize a loss function. Backpropagation may be used to calculate the gradient of the error function with respect to the neural network's weights.

When compliance and/or success in the model testing in step **614** is achieved, process flow proceeds to step **616**, where model deployment is triggered. The model may be utilized in AI functions and programming, for example to simulate intelligent behavior, to perform machine-assisted or computerized tasks, of which visual perception, speech recognition, decision-making, translation, forecasting, predictive modelling, and/or automated suggestion generation serve as non-limiting examples.

As discussed above, oversight of a deployed machine learning model may be automatically performed via a feedback loop whereby the method assesses performance of the deployed model (see step **616**) and the feedback loop automatically provides feedback for further training of the machine learning model to improve its performance, and upon completion of the other method steps such as **612**, the machine learning model that has been automatically retrained based on the feedback loop is then redeployed (step **614**). In some embodiments, the system is continually receiving training data as new predictions are made and more data is collected. The continuous training data may be discretized to generate input data to retrain the model. Discretization methods can convert continuous data to discrete data by binning, clustering, and numerical discretization. The model may monitor incoming data sets to make predictions. When predictions are made the system analyzes the predictions to determine whether the model needs to be retrained.

In some embodiments, the model may detect anomalies in the predictions. Anomaly detection can provide a benefit by identifying instances of the prediction that deviate from expected data or a general pattern. A difficulty in anomaly detection is that the system must define the boundary between ordinary data and anomalous data to accurately classify the data as ordinary or anomalous. The line between ordinary and anomalous may be difficult to determine with cases approaching a boundary and based on the specific application. For example, small variations may trigger an identification of an anomaly in the data while relatively larger deviations may be considered normal in less sensitive applications. The disclosed systems and methods may provide solutions to detecting anomalies in order to more accurately and quickly determine whether a model needs to be retrained. If data would be inapplicable or would corrupt the model by reducing the quality of the input data or training process (e.g., due to missing values, outliers, inconsistent formatting, incorrect labels, noisy data, etc.) that data may be automatically dropped and the source of that data may be blocked from providing data that would be used to train the model. This reflects an improvement in the process of training and deploying a model that is accurate and specific to the type of prediction sought. In particular, this provides an improvement in the field of model training, which provides a practical application.

In other applications, the anomaly detections processes described herein may be used to provide enhanced security to the overall computing system by detecting compliance gaps such as vulnerabilities in system security. For example, the system may take proactive measures to remediate danger by rectifying the one or more compliance gaps to remedy the security vulnerability. For example, the system may identify compliance gaps associated with encryption requirements and may remedy the encryption process to reduce the likelihood that data may be compromised, which provides an improvement in network security.

The systems and methods disclosed herein may also be used to analyze text to form the predictions. In particular, the systems and methods described herein include a combination of elements that are utilized in a specific manner for automatically performing automated processes based on technological efficiency, which provides a specific improvement over prior art systems resulting in improved computer processing for faster automated processing functions. For example, the systems and method may apply process automation for digital transformation of the data based on specific criteria to interpret text and unstructured data using text processing software techniques. The interpretation of the text may be implemented using the models described herein including unsupervised learning techniques or supervised learning techniques. The processor may track how much memory and/or processing time has been allocated to perform a function and the system may be trained to automatically detect and identify processes eligible for increased efficiencies based on existing inefficiencies in the process.

For example, the machine learning models may use unsupervised learning to identify and characterize hidden structures of unstructured and unlabeled content data, or supervised techniques that operate on labeled content data and include instructions informing the system which outputs are related to specific input values. In such instances, software processing can rely on iterative training techniques and training data to configure neural networks with an understanding of individual words, phrases, subjects, sentiments, and parts of speech.

Supervised learning software systems are trained using content data that is labeled or "tagged." During training, the supervised software systems learn the best mapping function between a known data input and expected known output

(i.e., labeled or tagged content data). Supervised natural language processing software then uses the best approximating mapping learned during training to analyze unforeseen input data (never seen before) to accurately predict the corresponding output. Supervised learning software systems often require extensive and iterative optimization cycles to adjust the input-output mapping until they converge to an expected and well-accepted level of performance, such as an acceptable threshold error rate between a calculated probability and a desired threshold probability.

The software systems are supervised because the way of learning from training data mimics the same process of a teacher supervising the end-to-end learning process. Supervised learning software systems are typically capable of achieving excellent levels of performance, but this excellent level of performance requires labeled data to be available. Developing, scaling, deploying, and maintaining accurate supervised learning software systems can take significant time, resources, and technical expertise from a team of skilled data scientists. Moreover, precision of the systems is dependent on the availability of labeled content data for training that is comparable to the corpus of content data that the system will process in a production environment.

Supervised learning software systems implement techniques that include, without limitation, Latent Semantic Analysis ("LSA"), Probabilistic Latent Semantic Analysis ("PLSA"), Latent Dirichlet Allocation ("LDA"), and more recent Bidirectional Encoder Representations from Transformers ("BERT"). Latent Semantic Analysis software processing techniques process a corporate of content data files to ascertain statistical co-occurrences of words that appear together, which then give insights into the subjects of those words and documents.

Unsupervised learning software systems can perform training operations on unlabeled data and less requirement for time and expertise from trained data scientists. Unsupervised learning software systems can be designed with integrated intelligence and automation to automatically discover information, structure, and patterns from content data. Unsupervised learning software systems can be implemented with clustering software techniques that include, without limitation, K-means clustering, Mean-Shift clustering, Density-based clustering, Spectral clustering, Principal Component Analysis, and Neural Topic Modeling ("NTM").

Clustering software techniques can automatically group semantically similar words together to accelerate the derivation and verification of an underneath common intent—i.e., ascertain or derive a new classification or subject, and not just classification into an existing subject or classification. Unsupervised learning software systems are also used for association rules mining to discover relationships between features from content data.

The content driver software service utilizes one or more supervised or unsupervised software processing techniques to perform a subject classification analysis to generate subject data. Suitable software processing techniques can include, without limitation, Latent Semantic Analysis, Probabilistic Latent Semantic Analysis, Latent Dirichlet Allocation. Latent Semantic Analysis software processing techniques generally process a corpus of alphanumeric text files, or documents, to ascertain statistical co-occurrences of words that appear together, which then give insights into the subjects of those words and documents. The content driver software service can utilize software processing techniques that include Non-Matrix Factorization, Correlated Topic Model ("CTM"), and K-Means or other types of clustering.

Neural networks may be trained using training set content data that comprise sample tokens, phrases, sentences, paragraphs, or documents for which desired subjects, content sources, interrogatories, or sentiment values are known. A labeling analysis may be performed on the training set content data to annotate the data with known subject labels, interrogatory labels, content source labels, or sentiment labels, thereby generating annotated training set content data. For example, a person can utilize a labeling software application to review training set content data to identify and tag or "annotate" various parts of speech, subjects, interrogatories, content sources, and sentiments.

The training set content data may then be fed to the content driver software service neural networks to identify subjects, content sources, or sentiments and the corresponding probabilities. For example, the analysis might identify that particular text represents a question with a 35% probability. If the annotations indicate the text is, in fact, a question, an error rate can be taken to be 65% or the difference between the calculated probability and the known certainty. Then parameters to the neural network are adjusted (i.e., constants and formulas that implement the nodes and connections between node), to increase the probability from 35% to ensure the neural network produces more accurate results, thereby reducing the error rate. The process is run iteratively on different sets of training set content data to continue to increase the accuracy of the neural network.

The content data is first pre-processes using a reduction analysis to create reduced content data. The reduction analysis first performs a qualification operation that removes unqualified content data that does not meaningfully contribute to the subject classification analysis. The qualification operation removes certain content data according to criteria defined by a provider. For instance, the qualification analysis can determine whether content data files are "empty" and contain no recorded linguistic interaction between a provider agent and a user and designate such empty files as not suitable for use in a subject classification analysis. As another example, the qualification analysis can designate files below a certain size below a given threshold (e.g., less than one minute) as also being unsuitable for use in the subject classification analysis.

The reduction analysis can also perform a contradiction operation to remove contradictions and punctuations from the content data. Contradictions and punctuation include removing or replacing abbreviated words or phrases that can cause inaccuracies in a subject classification analysis. Examples include removing or replacing the abbreviations "min" for minute, "u" for you, and "wanna" for "want to," as well as apparent misspellings, such as "mssed" for the word missed. In some embodiments, the contradictions can be replaced according to a standard library of known abbreviations, such as replacing the acronym "brb" with the phrase "be right back." The contradiction operation can also remove or replace contractions, such as replacing "we're" with "we are."

The reduction analysis can also streamline the content data by performing one or more of the following operations, including: (i) tokenization to transform the content data into a collection of words or key phrases having punctuation and capitalization removed; (ii) stop word removal where short, common words or phrases such as "the" or "is" are removed; (iii) lemmatization where words are transformed into a base form, like changing third person words to first person and changing past tense words to present tense; (iv) stemming to reduce words to a root form, such as changing plural to

singular; and (v) hyponymy and hypernym replacement where certain words are replaced with words having a similar meaning so as to reduce the variation of words within the content data.

Following a reduction analysis, the reduced content data is vectorized to map the alphanumeric text into a vector form. One approach to vectorizing content data includes applying "bag-of-words" modeling. The bag-of-words approach counts the number of times a particular word appears in content data to convert the words into a numerical value. The bag-of-words model can include parameters, such as setting a threshold on the number of times a word must appear to be included in the vectors.

Techniques to encode the context communication elements (e.g., such as words, speech patterns, tone, timbre, cadence, etc.) may, in part, determine how often communication elements appear together. Determining the adjacent pairing of communication elements can be achieved by creating a co-occurrence matrix with the value of each member of the matrix counting how frequently one communication element coincides with another, either just before or just after it. That is, the words or communication elements form the row and column labels of a matrix, and a numeric value appears in matrix elements that correspond to a row and column label for communication elements that appear adjacent in the content data.

As an alternative to counting communication elements (e.g., words) in a corpus of content data and turning it into a co-occurrence matrix, another software processing technique may be used where a communication element in the content data corpus predicts the next communication element. Looking through a corpus, counts may be generated for adjacent communication elements, and the counts are converted from frequencies into probabilities (i.e., using n-gram predictions with Kneser-Ney smoothing) using a simple neural network. Suitable neural network architectures for such purpose include a skip-gram architecture. The neural network may be trained by feeding through a large corpus of content data, and embedded middle layers in the neural network are adjusted to best predict the next word.

The predictive processing creates weight matrices that densely carry contextual, and hence semantic, information from the selected corpus of content data. Pre-trained, contextualized content data embedding can have high dimensionality. To reduce the dimensionality, a uniform manifold approximation and projection algorithm ("UMAP") can be applied to reduce dimensionality while maintaining essential information.

Prior to conducting a subject analysis to ascertain subject identifiers in the content data (i.e., topics or subjects addressed in the content data) or interaction driver identifiers in the content data (i.e., reasons why the customer initiated the interaction with the provider, such as the reason underlying a support request), the system can perform a concentration analysis on the content data. The concentration analysis concentrates, or increases the density of, the content data by identifying and retaining communication elements that have significant weight in the subject analysis and discarding or ignoring communication elements that have relativity little weight.

In one embodiment, the concentration analysis includes executing a term frequency-inverse document frequency ("tf-idf") software processing technique to determine the frequency or corresponding weight quantifier for communication elements with the content data. The weight quantifiers are compared against a pre-determined weight threshold to generate concentrated content data that is made up of communication elements having weight quantifiers above the weight threshold.

The concentrated content data is processed using a subject classification analysis to determine subject identifiers (i.e., topics) addressed within the content data. The subject classification analysis can specifically identify one or more interaction driver identifiers that are the reason why a user initiated a shared experience or support service request. An interaction driver identifier can be determined by, for example, first determining the subject identifiers having the highest weight quantifiers (e.g., frequencies or probabilities) and comparing such subject identifiers against a database of known interaction driver identifiers.

In one embodiment, the subject classification analysis is performed on the content data using a Latent Dirichlet Allocation analysis to identify subject data that includes one or more subject identifiers (e.g., topics addressed in the underlying content data). Performing the LDA analysis on the reduced content data may include transforming the content data into an array of text data representing key words or phrases that represent a subject (e.g., a bag-of-words array) and determining the one or more subjects through analysis of the array. Each cell in the array can represent the probability that given text data relates to a subject. A subject is then represented by a specified number of words or phrases having the highest probabilities (i.e., the words with the five highest probabilities), or the subject is represented by text data having probabilities above a predetermined subject probability threshold.

Clustering software processing techniques include K-means clustering, which is an unsupervised processing technique that does not utilized labeled content data. Clusters are defined by "K" number of centroids where each centroid is a point that represents the center of a cluster. The K-means processing technique run in an iterative fashion where each centroid is initially placed randomly in the vector space of the dataset, and the centroid moves to the center of the points that is closest to the centroid. In each new iteration, the distance between each centroid and the points are recalculated, and the centroid moves again to the center of the closest points. The processing completes when the position or the groups no longer change or when the distance in which the centroids change does not surpass a pre-defined threshold.

The clustering analysis yields a group of words or communication elements associated with each cluster, which can be referred to as subject vectors. Subjects may each include one or more subject vectors where each subject vector includes one or more identified communication elements (i.e., keywords, phrases, symbols, etc.) within the content data as well as a frequency of the one or more communication elements within the content data. The content driver software service can be configured to perform an additional concentration analysis following the clustering analysis that selects a pre-defined number of communication elements from each cluster to generate a descriptor set, such as the five or ten words having the highest weights in terms of frequency of appearance (or in terms of the probability that the words or phrases represent the true subject when neural networking architecture is used). In one embodiment, the descriptor sets were analyzed to determine if the reasons driving a customer support request were identified by the descriptor set subject identifiers.

The software model may be evaluated according to three categories, including a "good match" where the support request reason(s) are identified by the top words in the

subject vector (i.e., the words with the highest weight or frequency), a "moderate" match where the support request reason(s) are identified by the second tier of words in the subject vector (i.e., words six to ten), and a "poor" match where, for instance, the top words in a subject vector do not match or identify the reasons the support request was initiated.

Alternatively, instead of selecting a pre-determined number of communication elements, post-clustering concentration analysis can analyze the subject vectors to identify communication elements that are included in several subject vectors having a weight quantifier (e.g., a frequency) below a specified weight threshold level that are then removed from the subject vectors. In this manner, the subject vectors are refined to exclude content data less likely to be related to a given subject. To reduce an effect of spam, the subject vectors may be analyzed, such that if one subject vector is determined to include communication elements that are rarely used in other subject vectors, then the communication elements are marked as having a poor subject correlation and is removed from the subject vector.

In another embodiment, the concentration analysis is performed on unclassified content data by mapping the communication elements within the content data to integer values. The content data is thus turned into a bag-of-words that includes integer values and the number of times the integers occur in content data. The bag-of-words is turned into a unit vector, where all the occurrences are normalized to the overall length. The unit vector may be compared to other subject vectors produced from an analysis of content data by taking the dot product of the two-unit vectors. All the dot products for all vectors in a given subject are added together to provide a weighting quantifier or score for the given subject identifier, which is taken as subject weighting data. A similar analysis can be performed on vectors created through other processing, such as K-means clustering or techniques that generate vectors where each word in the vector is replaced with a probability that the word represents a subject identifier or request driver data.

In one example, text mapping may be applied to a data processing workflow to categorize the text to one or more topological vector spaces, wherein each respective vector space of the one or more topological vector spaces includes associated rule functions. A topological vector space is a vector space over a topological field that is provided with topological features of the text using underlying representations of words for text classification.

To illustrate generating subject weighting data, for any given subject there may be numerous subject vectors. Assume that for most of subject vectors, the dot product will be close to zero—even if the given content data addresses the subject at issue. Since there are some subjects with numerous subject vectors, there may be numerous small dot products that are added together to provide a significant score. Put another way, the particular subject is addressed consistently throughout a document, several documents, sessions of the content data, and the recurrence of the carries significant weight.

In another embodiment, a predetermined threshold may be applied where any dot product that has a value less than the threshold is ignored and only stronger dot products above the threshold are summed for the score. In another embodiment, this threshold may be empirically verified against a training data set to provide a more accurate subject analysis.

In another example, a number of subject identifiers may be substantially different, with some subjects having orders of magnitude fewer subject vectors than do other subjects. The weight scoring might significantly favor relatively unimportant subjects that occur frequently in the content data. To address this problem, a linear scaling on the dot product scoring based on the number of subject vectors may be applied. The result provides a correction to the score so that important but less common subjects are weighed more heavily.

Once all scores are calculated for all subjects, then subjects may be sorted, and the most probable subjects are returned. The resulting output provides an array of subjects and strengths. In another embodiment, hashes may be used to store the subject vectors to provide a simple lookup of text data (e.g., words and phrases) and strengths. The one or more subject vectors can be represented by hashes of words and strengths, or alternatively an ordered byte stream (e.g., an ordered byte stream of 4-byte integers, etc.) with another array of strengths (e.g., 4-byte floating-point strengths, etc.).

The content driver software service can also use term frequency-inverse document frequency software processing techniques to vectorize the content data and generating weighting data that weight words or particular subjects. The tf-idf is represented by a statistical value that increases proportionally to the number of times a word appears in the content data. This frequency is offset by the number of separate content data instances that contain the word, which adjusts for the fact that some words appear more frequently in general across multiple shared experiences or content data files. The result is a weight in favor of words or terms more likely to be important within the content data, which in turn can be used to weigh some subjects more heavily in importance than others. To illustrate with a simplified example, the tf-idf might indicate that the term "password" carries significant weight within content data. To the extent any of the subjects identified by a natural language processing analysis include the term "password," that subject can be assigned more weight by the content driver software service.

The content data can be visualized and subject to a reduction into two-dimensional data using a UMAP to generate a cluster graph visualizing a plurality of clusters. The content driver software service feeds the two-dimensional data into a DBSCAN and identify a center of each cluster of the plurality of clusters. The process may, using the two-dimensional data from the UMAP and the center of each cluster from the DBSCAN, apply a KNN to identify data points closest to the center of each cluster and shade each of the data points to graphically identify each cluster of the plurality of clusters. The processor may illustrate a graph on the display representative of the data points that are shaded following application of the KNN.

The content driver software service can also incorporate Part of Speech ("POS") tagging software code that assigns words a part of speech depending upon the neighboring words, such as tagging words as a noun, pronoun, verb, adverb, adjective, conjunction, preposition, or other relevant parts of speech. The content driver software service can utilize the POS tagged words to help identify questions and subjects according to pre-defined rules, such as recognizing that the word "what" followed by a verb is also more likely to be a question than the word "what" followed by a preposition or pronoun (e.g., "What is this?" versus "What he wants is an answer.").

POS tagging in conjunction with Named Entity Recognition ("NER") software processing techniques can be used by the content driver software service to identify various content sources within the content data. NER techniques are utilized to classify a given word into a category, such as a

person, product, organization, or location. Using POS and NER techniques to process the content data allow the content driver software service to identify particular words and text as a noun and as representing a person participating in the discussion (e.g., a content source).

The systems and methods disclosed herein may utilize deployed models (i.e., machine learning models, neural networks, predictive models, etc.) to make predictions about text in order to perform software control. The use of specially trained models realizes a number of improvements over traditional methods of software controls, including more accurate and uniform compliance with respect to a predefined risk posture of an entity and government regulations through providing uniform controls and review processes. Advantageously, the systems improve error detection associated with non-compliance. Further, the systems and methods disclosed herein lead to faster training times and a more accurate model.

The systems and methods disclosed herein reflect an improvement in the functioning of a computer or an improvement to other technology or a technical field by enhancing data retention and destruction systems. Further, the systems and methods disclosed herein utilize a combination of software and hardware that include, for example, a physical circuit, which is a machine or manufacture. The systems and methods disclosed herein provide an improvement to data retention and destruction processes, which can lead to improved compliance with governmental regulations and reduce security vulnerabilities. In the context of a financial institution, the systems and methods enhance existing data governance processes associated with governmental regulations related to retention of financial records.

Advantageously, the processes disclosed herein provide more consistency across departments in an organization related to data destruction and retention. Specifically, the disclosed systems and methods reduce the amount of outdated or irrelevant data that is being unnecessarily stored, which can reduce the amount of data being stored to data storage locations, which can improve computing efficiencies and computing functionality as it reduces the time it takes for the computing system to locate important data. By appropriately destroying data that is no longer permitted to be retained, this can reduce the legal exposure of an organization. The systems and methods disclosed herein can reduce data retention errors that are associated with manual destruction and inconsistent adherence to governmental regulations and organizational policies. The systems and methods disclosed herein relate to database and file management and involve data retention and destruction processes, which is specifically directed to a technical improvement arising in the realm of computer memory and data storage processes. The system controls access to the data destruction workflow, can perform authentication review of credentials to verify the individual has authority to destroy data, there are multiple checks by communicating across a network with multiple remote individuals including a solution owner and a retention lead prior to destroying data. This non-generic combination of elements for implementing the data retention and destruction workflow results in an improvement in the technology of implementing records management because it eliminates manual error and provides greater consistency across disparate servers, databases, and various data storage locations for many different applications and questionnaires. Questionnaires may include questions associated with risk assessment for extracting actionable insights, engaging stakeholders, and managing identified risks.

FIG. **7** depicts an example architecture for data retention and destruction, according to an embodiment of the present invention. The architecture includes module inventory **700**A that includes metadata to an inventory of modules, where the modules identify records of applications and questionnaires **700**B. The module inventory **700**A is in communication with data classification inventory **700**D, which is inventory of data classifications and retention periods. The architecture is part of a records management workflow where a fetch application programming interface (API) **700**C is used to access the module inventory **700**A to obtain the scope of the records that are subject to destruction. Records that are subject to destruction would be records that have been in existence beyond the retention periods specified by the data classification inventory **700**D. Specifically, when a person logs in, they obtain a session token. The fetch API **700**C then obtains module metadata of the modules from the module inventory **700**A and executes a custom extensible markup language (XML) search to find records in the applications and questionnaires **700**B. Further, the fetch API is used to create, update, or otherwise obtain snapshots to capture historical data of the inventory modules from the records management protocol **700**E. A document destruction API **700**F is then implemented that obtains authorized snapshots from the records management protocol **700**E to identify target records within the applications and questionnaires **700**B that are eligible for destruction. The document destruction API **700**F loops through the records results and deletes the target records. In addition snapshot record data is cleared.

FIGS. **8-9** depict module inventory information and a front-end record setup interface **800** associated with the module inventory as well as a front-end interface **900** for a date field used for data retention that is associated with the module inventory. Specifically, the module inventory information indicates information about the records of the module, which can include the module name, the module status, the application owner, the retention lead, the data field, the data classification, the retention period, the module identification number, and the date field identification number. FIG. **10** depicts additional information **1000** included in the front-end interface of FIG. **9** for specifying a retention lead. A front-end interface **1100** for confirming enrollment associated with a data retention and destruction settings is depicted by FIG. **11** and FIG. **12** depicts a front-end interface **1200** for administrative setup associated with the module inventory. FIG. **13** depicts a front-end interface **1300** associated with retention lead and fields to display that is associated with the module inventory.

In some embodiments, the machine learning workflow described by FIG. **6** may be used to predict information that should be included in the data classification. For example, the machine learning workflow of FIG. **6** may predict whether data is subject to certain governmental regulations and/or organizational policies in order assign retention periods to the data.

FIG. **14** depicts data classification inventory information **1400**, in accordance with an embodiment of the present invention. The data classification inventory information can indicate a classification, a category, a retention period (textual description) and retention period days (number). FIG. **15** depicts a front-end interface **1500** for selecting data classification associated with the data classification inventory.

FIG. **16** depicts records management information **1600** associated with records management. The records management information **1600** can indicate the record identification

number, the module name, the date field value, whether the record should be held, the reason the record should be held, the date the record is placed on hold, the date to revisit the hold, the module identification number, and the content identification number. FIG. **17** depicts a front-end interface **1700** depicting records pending decision that is associated with records management, and FIG. **18** depicts a front-end interface **1800** depicting information necessary for placing a record on hold, which is associated with the records management protocol.

FIG. **19** depicts a front-end interface **1900** depicting records that are on hold, and FIG. **20** depicts a front-end interface for authorizing destruction of records, such as records that are on hold and not programmed for automatic destruction. FIG. **21** depicts a front-end interface **2100** for logging destroyed records.

FIG. **22** depicts an example data classification workflow facilitating records management. Initially, the application inventory **2200**A is enrolled by the solution owner **2200**B and a retention lead sets up the data document and retention policies **2200**C. The system may then implement the automated data retention and destruction processes described herein which implement a first API **2200**D to get the scope of the records to be destroyed and finds the records. A retention lead then reviews any records that are subject to hold **2200**E, and a retention lead authorizes the data to be destroyed **2200**F. A second API then obtains the records that are authorized for destruction and deletes the records **2200**G. FIG. **23** depicts a front-end interface **2300** associated with tag snapshots of records that have been locked and authorized for destruction.

FIG. **24** depicts a front-end interface **2400** for data retention and destruction reporting. Specifically, the front-end interface indicates the number of enrolled applications, the applications that are not enrolled in the data retention and destruction workflow. The front-end interface **2400** indicates the number of records for destruction that are pending confirmation and the number of records that are pending configuration. In addition, the front-end interface **2400** indicates the records that are authorized for destruction across all modules and the total records on hold across all modules. FIG. **25** depicts a front-end interface **2500** for accessing reports associated with data retention and destruction, and FIG. **26** depicts a front-end interface **2600** associated with a data retention and destruction repository. Further, FIG. **27** depicts an example report **2700** associated with data retention and destruction, and FIG. **28** depicts an example report **2800** of destroyed records.

FIG. **29** depicts a block diagram of an example method **2900**, in accordance with an embodiment of the present invention. At block **2905**, the method includes assigning, via a data classification workflow, metadata to an inventory of modules. In some embodiments, the inventory of the modules includes information that includes a module name, a module status indicating a list of values, an indication of an application owner of one or more records, a name of a retention lead from which authorization to destroy the one or more records is to be obtained, a date field, a data classification, a retention period, a module identification number, and a date field identification number. In some embodiments, the data classification workflow is controlled by an information rights management (IRM) security restriction mechanism. In some embodiment, the data classification workflow incorporates governmental regulations for records retention that apply to a financial entity and corporate data retention policies.

At block **2910**, the system processes, via a records management workflow, a fetch application programming interface (API) to access the inventory of the modules, the fetch API including (i) obtaining module metadata of the modules from the inventory, (ii) executing an extensible markup language (XML) search for applications and the questionnaires, and (iii) obtaining snapshots to capture historical data of the modules.

At block **2915**, the system implements, via a document destruction API, the records management protocol to identify target records within the applications and the questionnaires that are eligible for destruction. The records management protocol may utilize, in some embodiments, the metadata that is assigned to records to each of the modules to identify the target records, the metadata including a record identification number, a module name, a date field value, a hold value, a hold reason, a date placed on hold, a date to revisit, a module identification number, and a content identification number. The records management protocol includes distributing the snapshots of the historical data indicating the modules that are eligible for the destruction, scanning the modules for the target records, and deleting the target records. In some embodiments, the records management protocol utilizes data classification information and retention information and the target records that are eligible for the destruction are identified based on the data classification information and the retention information, the data classification information and the retention information being established using the records management workflow. For example, the target records that are identified are those records that have been in existence beyond a predefined retention period specified by the retention information. In addition, the system populates a log of destruction actions with an entry indicating the target records have been deleted.

In some embodiments, the system records management workflow further includes obtaining confirmation to delete the target records is required prior to the scanning of the modules and the deleting of the target records, particularly if the records have been designated for a record hold. In some embodiments, the system tags the snapshots with an authorization data when authorization to delete the target records was obtained. In some embodiments, the method includes obtaining a hold request to retain one or more of the records of the applications and the questionnaires. In some embodiments, hold details of the hold request are processed and metadata in the inventory of the modules is updated with the hold details. In some embodiments, the hold details indicate (a) a reason for retaining the one or more records, (b) a date to revisit the one or more records, (c) a justification for the reason for the hold, and (d) a reference number.

FIG. **30** depicts a block diagram of an example method **3000**, in accordance with an embodiment of the present invention. At block **3005**, the system authenticates authentication credentials against stored credentials of a secure credential repository and based thereon enabling a computing device to access a computer workflow engine. At block **3010**, the system executes, in response to one or more inputs from the computing device, a data classification workflow, the data classification workflow assigning metadata to an inventory of modules. At block **3015**, the system processes, via a records management workflow, a fetch application programming interface (API) to access the inventory of the modules, the fetch API including (i) obtaining module metadata of the modules from the inventory, (ii) executing an extensible markup language (XML) search for applications and the questionnaires, and (iii) obtaining snapshots to capture historical data of the modules. At block **3020**, the

system implements, via a document destruction API, the records management protocol to identify target records within the applications and the questionnaires that are eligible for destruction. The records management protocol includes distributing the snapshots of the historical data indicating the modules that are eligible for the destruction, scanning the modules for the target records, and deleting the target records.

In some embodiments, the records management protocol utilizes data classification information and retention information and the target records that are eligible for the destruction are identified based on the data classification information and the retention information, the data classification information and the retention information being established using the records management workflow. In some embodiments, the target records that are identified are in existence beyond a predefined retention period specified by the retention information.

FIG. **31** depicts a block diagram of an example method **3100** for implementing a records management protocol, in accordance with an embodiment of the present invention. At block **3105**, the system initiates display of a module inventory interface, the module inventory interface including one or more inputs for data retention and destruction enrollment, at least one input of the one or more inputs configured to define a retention period for one or more objects. At block **3110**, the system receives an indication of a defined retention period for an object, the indication being received based on the one or more inputs being used to indicate the defined retention period for the object. At block **3115**, the system initiates display, via the module inventory interface, a required input for defining a field to be used for the defined retention period for the object. At block **3120**, the system receives a selection for the field to be used for the defined retention period for the object. At block **3125**, the system initiates display, via the module inventory interface, of a retention lead input for indicating a records reviewer for authorizing destruction of the object outside of the defined retention period or retaining the object beyond the defined retention period.

At block **3130**, the system receives a name for the records reviewer based on the name being provided via the retention lead input. At block **3135**, the system initiates display of a confirmation input for receiving confirmation of selections received via the module inventory interface, and at block **3140** the system receives the confirmation of the selections provided via the confirmation input.

In some embodiments, the confirmation of the selections authorizes the object to automatically be permanently deleted upon completion of the defined retention period. In some embodiments, the object includes data records associated with an application or a questionnaire. In some embodiments, the system initiates display, via the module inventory interface, of a confirmation page indicating the data retention and destruction enrollment is complete and that implementation of data retention and destruction parameters are configurable. Further, the system may initiate display, via the module inventory interface, of a records summary indicating records pending a decision and records that are on-hold. In some embodiments, the system initiates display, via a records management interface, of a table with detailed analysis of the records pending the decision. In addition, the table may indicate a respective data record number for each of the records, respective links to each of the records, policy holder information for each of the records, date fields to be used as a data retention marker for each of the records, and an update date for when each of the records were updated.

In some embodiments, the system initiates display of a hold details enrollment section for providing details about the records that are to be enrolled for a retention hold, the hold details enrollment section including one or more control inputs for indicating: (a) a hold reason, (b) a date the records are being placed on hold, (c) a hold justification, and (d) a date to revisit the retention hold.

The system can also initiate display of a data classification inventory interface to define the data retention and destruction parameters. Further, the data retention and destruction parameters may indicate one or more relevant governmental regulations applicable to the object that dictate the defined retention period. The data retention and destruction parameters may, in some embodiments, indicate one or more relevant organizational policies applicable to the object.

In some embodiments, the system initiates display of a destruction authorization interface for providing authorization to automatically delete a list of records. In some instances, the system may initiate display of a log of destroyed records, the log of destroyed records indicating the date the destroyed records were destroyed. In some examples, the system may initiate display of an enrollment report that includes the object enrolled by the selections received via the module inventory interface. Further, the enrollment report may further indicate records that (a) have not yet been enrolled, (b) are pending confirmation, (c) are pending configuration, (d) are authorized for destruction, and (e) are on hold. In some embodiments, the system assigns metadata to the object, the metadata indicating the selections. In some embodiments, access to the module inventory interface is restricted in accordance with an information rights management (IRM) security restriction mechanism. In some embodiments, the system tags snapshots of the selections received via the module inventory interface.

FIG. **32** depicts a block diagram of an example method **3200**, in accordance with an embodiment of the present invention. At block **3205**, the system initiates display of a module inventory interface, the module inventory interface including one or more inputs for data retention and destruction enrollment, at least one input of the one or more inputs configured to define a retention period for one or more objects. At block **3210**, the system receives an indication of a defined retention period for an object, the indication being received based on the one or more inputs being used to indicate the defined retention period for the object. At block **3215**, the system initiates display, via the module inventory interface, a required input for defining a field to be used for the defined retention period for the object. At block **3220**, the system receives a selection for the field to be used for the defined retention period for the object. At block **3225**, the system initiates display, via the module inventory interface, of (i) a retention lead input for indicating a records reviewer for authorizing destruction of the object outside of the defined retention period or retaining the object beyond the defined retention period, and (ii) a plurality of contextual fields to provide context to the object. At block **3230**, the system receives a name for the records reviewer based on the name being provided via the retention lead input. At block **3235**, the system initiates display of a confirmation input for receiving confirmation of selections received via the module inventory interface, and at block **3240**, the system receives the confirmation of the selections provided via the confirmation input.

FIGS. **33**A-**33**B depict a block diagram of an example method **3300**, in accordance with an embodiment of the present invention. At block **3305**, the system initiates display of a module inventory interface, the module inventory interface including one or more inputs for data retention and destruction enrollment, at least one input of the one or more inputs configured to define a retention period for one or more records. At block **3310**, the system receives an indication of a defined retention period for a record of the one or more records, the indication being received based on the one or more inputs being used to indicate the defined retention period for the record. At block **3315**, the system initiates display, via the module inventory interface, a required input for defining a field to be used for the defined retention period for the record. At block **3320**, the system receives a selection for the field to be used for the defined retention period for the record. At block **3325**, the system initiates display, via the module inventory interface, of a retention lead input for indicating a records reviewer for authorizing destruction of the record outside of the defined retention period or retaining the object beyond the defined retention period. At block **3330**, the system receives a name for the records reviewer based on the name being provided via the retention lead input. At block **3335**, the system initiates display of a confirmation input for receiving confirmation of selections received via the module inventory interface. At block **3340**, the system receives the confirmation of the selections provided via the confirmation input. At block **3340**, the system assigns retention-related metadata to the record. At block **3345**, the system processes a fetch application programming interface (API) to access a records inventory, the fetch API including (i) obtaining module metadata of the one or more records from the records inventory, (ii) executing an extensible markup language (XML) search for the one or more records, and (iii) obtaining snapshots to capture historical data of the one or more records. At block **3345**, the system implements, via a document destruction API, a records management protocol to identify target records within the one or more records that are eligible for destruction. The records management protocol includes distributing the snapshots of the historical data indicating the target records that are eligible for the destruction, scanning the records inventory for the target records, and deleting the target records.

FIGS. **34**A-**34**B depict a block diagram of an example method **3400**, in accordance with an embodiment of the present invention. At block **3405**, the system initiates display of a module inventory interface, the module inventory interface including one or more inputs for data retention and destruction enrollment, at least one input of the one or more inputs configured to define a retention period for one or more records. At block **3410**, the system receives an indication of a defined retention period for a record of the one or more records, the indication being received based on the one or more inputs being used to indicate the defined retention period for the record. At block **3415**, the system initiates display, via the module inventory interface, a required input for defining a field to be used for the defined retention period for the record. At block **3420**, the system receives a selection for the field to be used for the defined retention period for the record. At block **3425**, the system initiates display, via the module inventory interface, of (i) a retention lead input for indicating a records reviewer for authorizing destruction of the record outside of the defined retention period or retaining the record beyond the defined retention period, and (ii) a plurality of contextual fields to provide context to the record. At block **3430**, the system receives a name for the records reviewer based on the name being provided via the retention lead input. At block **3435**, the system initiates display of a confirmation input for receiving confirmation of selections received via the module inventory interface. At block **3440**, the system receives the confirmation of the selections provided via the confirmation input. At block **3445**, the system assigns retention-related metadata to the record. At block **3450**, the system processes a fetch application programming interface (API) to access a records inventory, the fetch API including (i) obtaining module metadata of the one or more records from the records inventory, (ii) executing an extensible markup language (XML) search for the one or more records, and (iii) obtaining snapshots to capture historical data of the one or more records. At block **3455**, the system implements, via a document destruction API, a records management protocol to identify target records within the one or more records that are eligible for destruction. The records management protocol includes, distributing the snapshots of the historical data indicating the target records that are eligible for the destruction, scanning the records inventory for the target records, and deleting the target records.

Aspects of the present invention are described herein with reference to flowchart illustrations and/or block diagrams of computer-implemented methods and computing systems according to embodiments of the invention. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer readable program instructions that may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus (the term "apparatus" includes systems and computer program products). The processor may execute the computer readable program instructions thereby creating a means for implementing the actions specified in the flowchart illustrations and/or block diagrams. These computer readable program instructions may also be stored in a computer readable storage medium that can direct a computer, a programmable data processing apparatus, and/or other devices to function in a particular manner, such that the computer readable storage medium having instructions stored therein comprises an article of manufacture including instructions which implement aspects of the actions specified in the flowchart illustrations and/or block diagrams. In particular, the computer readable program instructions may be used to produce a computer-implemented method by executing the instructions to implement the actions specified in the flowchart illustrations and/or block diagrams.

The computer program instructions may also be stored in a computer-readable memory that can direct a computer or other programmable data processing apparatus to function in a particular manner, such that the instructions stored in the computer readable memory produce an article of manufacture including instructions, which implement the function/act specified in the flowchart and/or block diagram block or blocks.

The computer program instructions may also be loaded onto a computer or other programmable data processing apparatus to cause a series of operational steps to be performed on the computer or other programmable apparatus to produce a computer implemented process such that the instructions, which execute on the computer or other programmable apparatus, provide steps for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks. Alternatively, computer program implemented steps or acts may be combined with operator or

human implemented steps or acts in order to carry out an embodiment of the invention.

In the flowchart illustrations and/or block diagrams disclosed herein, each block in the flowchart/diagrams may represent a module, segment, or portion of instructions, which comprises one or more executable instructions for implementing the specified logical function(s). In some implementations, the functions noted in the blocks may occur out of the order noted in the Figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved.

Computer program instructions are configured to carry out operations of the present invention and may be or may incorporate assembler instructions, instruction-set-architecture (ISA) instructions, machine instructions, machine dependent instructions, microcode, firmware instructions, state-setting data, source code, and/or object code written in any combination of one or more programming languages.

An application program may be deployed by providing computer infrastructure operable to perform one or more embodiments disclosed herein by integrating computer readable code into a computing system thereby performing the computer-implemented methods disclosed herein.

Although various computing environments are described above, these are only examples that can be used to incorporate and use one or more embodiments. Many variations are possible.

The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of the invention. As used herein, the singular forms "a," "an" and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms "comprise" (and any form of comprise, such as "comprises" and "comprising"), "have" (and any form of have, such as "has" and "having"), "include" (and any form of include, such as "includes" and "including"), and "contain" (and any form contain, such as "contains" and "containing") are open-ended linking verbs. As a result, a method or device that "comprises," "has," "includes," or "contains" one or more steps or elements possesses those one or more steps or elements but is not limited to possessing only those one or more steps or elements. Likewise, a step of a method or an element of a device that "comprises," "has," "includes," or "contains" one or more features possesses those one or more features but is not limited to possessing only those one or more features. Furthermore, a device or structure that is configured in a certain way is configured in at least that way but may also be configured in ways that are not listed.

The corresponding structures, materials, acts, and equivalents of all means or step plus function elements in the claims below, if any, are intended to include any structure, material, or act for performing the function in combination with other claimed elements as specifically claimed. The description of the present invention has been presented for purposes of illustration and description but is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art without departing from the scope and spirit of the invention. The embodiment was chosen and described in order to best explain the principles of one or more aspects of the invention and the practical application, and to enable others of ordinary skill in the art to understand one or more aspects of the invention for various embodiments with various modifications as are suited to the particular use contemplated.

What is claimed is:

1. A computing system for implementing a records management protocol controlling data retention and destruction, the system comprising:

at least one processor;

a communication interface communicatively coupled to the at least one processor; and

a memory device storing executable code that, when executed, causes the at least one processor to:

initiate display of a module inventory interface, the module inventory interface including one or more inputs for data retention and destruction enrollment, at least one input of the one or more inputs configured to define a retention period for one or more objects;

receive an indication of a defined retention period for an object, the indication being received based on the one or more inputs being used to indicate the defined retention period for the object;

initiate display, via the module inventory interface, a required input for defining a field to be used for the defined retention period for the object;

receive a selection for the field to be used for the defined retention period for the object;

initiate display, via the module inventory interface, of a retention lead input for indicating a records reviewer for authorizing destruction of the object outside of the defined retention period or retaining the object beyond the defined retention period;

receive a name for the records reviewer based on the name being provided via the retention lead input;

initiate display of a confirmation input for receiving confirmation of selections received via the module inventory interface;

receive the confirmation of the selections provided via the confirmation input;

capture one or more snapshots of historical data of the object based on the confirmation of the selections, and tag the one or more snapshots with authorization data that includes a name of a records reviewer with authorization to delete one or more target records; and

identify, based on the one or more tagged snapshots and the defined retention period, the one or more target records eligible for destruction, and delete the one or more target records.

2. The system of claim **1**, wherein the confirmation of the selections authorizes the object to automatically be permanently deleted upon completion of the defined retention period.

3. The system of claim **1**, wherein the object is associated with an application or a questionnaire.

4. The system of claim **1**, wherein the executable code, when executed, further causes the at least one processor to initiate display, via the module inventory interface, of a confirmation page indicating the data retention and destruction enrollment is complete and that implementation of data retention and destruction parameters are configurable.

5. The system of claim **4**, wherein the executable code, when executed, further causes the at least one processor to initiate display, via the module inventory interface, of a records summary indicating records pending a decision and records that are on-hold.

6. The system of claim **5**, wherein the wherein the executable code, when executed, further causes the at least

one processor to initiate display, via a records management interface, of a table with detailed analysis of the records pending the decision.

7. The system of claim **6**, wherein the table indicates a respective data record number for each of the records, respective links to each of the records, policy holder information for each of the records, date fields to be used as a data retention marker for each of the records, and an update date for when each of the records were updated.

8. The system of claim **5**, wherein the executable code, when executed, further causes the at least one processor to initiate display of hold details enrollment section for providing details about the records that are to be enrolled for a retention hold, the hold details enrollment section including one or more control inputs for indicating: (a) a hold reason, (b) a date the records are being placed on hold, (c) a hold justification, and (d) a date to revisit the retention hold.

9. The system of claim **4**, wherein the wherein the executable code, when executed, further causes the at least one processor to initiate display of a data classification inventory interface to define the data retention and destruction parameters.

10. The system of claim **9**, wherein the data retention and destruction parameters indicate one or more relevant governmental regulations applicable to the object that dictate the defined retention period.

11. The system of claim **9**, wherein the data retention and destruction parameters indicate one or more relevant organizational policies applicable to the object.

12. The system of claim **1**, wherein the executable code, when executed, further causes the at least one processor to initiate display of a destruction authorization interface for providing authorization to automatically delete a list of records.

13. The system of claim **1**, wherein the executable code, when executed, further causes the at least one processor to initiate display of a log of destroyed records, the log of destroyed records indicating the date the destroyed records were destroyed.

14. The system of claim **1**, wherein the executable code, when executed, further causes the at least one processor to initiate display of an enrollment report that includes the object enrolled by the selections received via the module inventory interface.

15. The system of claim **14**, wherein the enrollment report further indicates records that (a) have not yet been enrolled, (b) are pending confirmation, (c) are pending configuration, (d) are authorized for destruction, and (e) are on hold.

16. The system of claim **1**, wherein the executable code, when executed, further causes the at least one processor to assign metadata to the object, the metadata indicating the selections.

17. The system of claim **1**, wherein access to the module inventory interface is restricted in accordance with an information rights management (IRM) security restriction mechanism.

18. The system of claim **1**, wherein the executable code, when executed, further causes the at least one processor to tag snapshots of the selections received via the module inventory interface.

19. A computing system, comprising:

at least one processor;

a communication interface communicatively coupled to the at least one processor; and

a memory device storing executable code that, when executed, causes the at least one processor to:

initiate display of a module inventory interface, the module inventory interface including one or more inputs for data retention and destruction enrollment, at least one input of the one or more inputs configured to define a retention period for one or more objects;

receive an indication of a defined retention period for an object, the indication being received based on the one or more inputs being used to indicate the defined retention period for the object;

initiate display, via the module inventory interface, a required input for defining a field to be used for the defined retention period for the object;

receive a selection for the field to be used for the defined retention period for the object;

initiate display, via the module inventory interface, of (i) a retention lead input for indicating a records reviewer for authorizing destruction of the object outside of the defined retention period or retaining the object beyond the defined retention period, and (ii) a plurality of contextual fields to provide context to the object;

receive a name for the records reviewer based on the name being provided via the retention lead input;

initiate display of a confirmation input for receiving confirmation of selections received via the module inventory interface;

receive the confirmation of the selections provided via the confirmation input;

capture one or more snapshots of historical data of the object based on the confirmation of the selections, and tag the one or more snapshots with authorization data that includes a name of a records reviewer with authorization to delete one or more target records; and

identify, based on the one or more tagged snapshots and the defined retention period, the one or more target records eligible for destruction, and delete the one or more target records.

20. A computer-implemented method, comprising:

initiating display of a module inventory interface, the module inventory interface including one or more inputs for data retention and destruction enrollment, at least one input of the one or more inputs configured to define a retention period for one or more objects;

receiving an indication of a defined retention period for an object, the indication being received based on the one or more inputs being used to indicate the defined retention period for the object;

initiating display, via the module inventory interface, a required input for defining a field to be used for the defined retention period for the object;

receiving a selection for the field to be used for the defined retention period for the object;

initiating display, via the module inventory interface, of a retention lead input for indicating a records reviewer for authorizing destruction of the object outside of the defined retention period or retaining the object beyond the defined retention period;

receiving a name for the records reviewer based on the name being provided via the retention lead input;

initiating display of a confirmation input for receiving confirmation of selections received via the module inventory interface;

receiving the confirmation of the selections provided via the confirmation input;

capturing one or more snapshots of historical data of the object based on the confirmation of the selections, and tagging the one or more snapshots with authorization data that includes a name of a records reviewer with authorization to delete one or more target records; and

identifying, based on the one or more tagged snapshots and the defined retention period, the one or more target records eligible for destruction, and deleting the one or more target records.

* * * * *